IBM

Appliance Control Center for IBM Z and LinuxONE User's Guide

GC28-7073-00

Note:

Before you use this information and the product it supports, read the information in "Safety" on page vii, "Notices" on page 115, and *IBM Systems Environmental Notices and User Guide*, Z125–5823.

This edition, GC28-7073-00, applies to IBM z17 (Model ME1) and IBM LinuxONE Emperor 5 (Model ML1).

There might be a newer version of this document in a **PDF** file available on **IBM Documentation**. Go to https://www.ibm.com/docs/en/systems-hardware, select **IBM Z** or **IBM LinuxONE**, then select your configuration, and click **Library Overview** on the navigation bar.

Contents

Safety....................vii
Safety notices.................... vii
World trade safety information....................vii
Laser safety information....................vii
Laser compliance.................... vii

About this publication....................ix
Related publications....................ix
Related HMC and SE console information.................... ix
Accessibility features.................... ix
Consult assistive technologies.................... ix
Keyboard navigation.................... ix
IBM and accessibility.................... ix
How to provide feedback to IBM....................ix

Summary of changes....................xi

Chapter 1. Introduction to Appliance Control Center for IBM Z & IBM LinuxONE...... 1
Solution overview....................1
Modes of Operation....................1
Connectivity considerations.................... 3
Limitations....................3
Planning for Appliance Control Center for IBM Z & IBM LinuxONE....................3
Securing Appliance and Appliance Control Center.................... 6
Verifying Image Integrity....................7
Credentials used by Appliance Control Center....................8

Chapter 2. Installing Appliance Control Center for IBM Z & IBM LinuxONE............11
Checklist....................11
Downloading Appliance Control Center from Fix Central....................12
Installing Appliance Control Center for IBM Z & IBM LinuxONE using Ansible playbook....................13
Installing Appliance Control Center for IBM Z & IBM LinuxONE using HMC.................... 14

Chapter 3. Configuring Appliance Control Center for IBM Z & IBM LinuxONE..........17
Terminologies....................17
Configuring Appliance Control Center for IBM Z & IBM LinuxONE using Ansible playbook....................18
Configuring Appliance Control Center for IBM Z & IBM LinuxONE using the APIs....................19
Configuring Multi-factor Authentication....................25
Enabling Two-Factor Authentication using Ansible Playbook.................... 26
Enabling two-factor authentication using APIs....................27
Generating MFA Secrets for Admins and Appliance Owners.................... 27
Disabling two-factor authentication.................... 28
Using CA-Signed Certificates....................28
Generating CSR in ACC....................28
Uploading the CA-Signed Certificate to ACC.................... 29
Getting Logs of Appliance Control Center.................... 29
Gathering logs using Ansible Playbook....................29
Gathering logs manually.................... 30
Upgrading Appliance Control Center for IBM Z & IBM LinuxONE.................... 31
Creating an ACC Restore Checkpoint.................... 31

Chapter 4. Installing an Appliance using Appliance Control Center for IBM Z & IBM LinuxONE.... 33
Install Appliance using Ansible Playbooks.... 33
Installing an Appliance using API commands.... 34
Installing Appliances using ACC UI.... 37

Chapter 5. Activating and Deactivating an appliance.... 45
Activating an Appliance using API.... 45
Deactivating an Appliance using API.... 45
Activating an Appliance using UI.... 46
Deactivating Appliance using UI.... 46

Chapter 6. Unlocking the Appliance 47
Unlocking Appliances using API.... 47
Unlocking Appliance using UI.... 48

Chapter 7. Health Monitoring.... 51
Health Monitoring Using API.... 51
Health Monitoring Using UI.... 52

Chapter 8. Getting Logs of Appliances using Appliance Control Center.... 53
Generating an Authentication Token.... 53
Creating Appliance Logs Using ACC Alerts API.... 53
Retrieving a List of Alerts.... 55
Downloading Logs from Appliances.... 55

Chapter 9. Updating Appliances using Appliance Control Center for IBM Z & IBM LinuxONE 57
Checklist.... 57
Updating Appliances by using Ansible Playbook 57
Upgrading Appliances as a Cluster.... 58
Updating Appliance by using an Image.... 58
Rolling Back an Appliance Update 60

Chapter 10. API Reference.... 61
Initializing the Appliance Control Center (ACC).... 61
User Management.... 62
Multi-Factor Authentication.... 65
Certificate management.... 67
Configuration.... 68
CPCs Management.... 69
Sync in between ACC and HMC.... 71
Resource Package.... 72
Managing Appliance Images.... 78
Managing Appliance Cluster 80
Background tasks.... 86
Alerts.... 86
Appliance Restart.... 91

Chapter 11. Troubleshooting.... 93
Messages.... 96

Notices.... 115
Trademarks.... 115

Class A Notices.. 116

Safety

Safety notices

Safety notices may be printed throughout this guide. **DANGER** notices warn you of conditions or procedures that can result in death or severe personal injury. **CAUTION** notices warn you of conditions or procedures that can cause personal injury that is neither lethal nor extremely hazardous. **Attention** notices warn you of conditions or procedures that can cause damage to machines, equipment, or programs.

World trade safety information

Several countries require the safety information contained in product publications to be provided in their local language(s). If this requirement applies to your country, a safety information booklet is included in the publications package shipped with the product. The booklet contains the translated safety information with references to the US English source. Before using a US English publication to install, operate, or service this product, you must first become familiar with the related safety information in the *Systems Safety Notices*, G229-9054. You should also refer to the booklet any time you do not clearly understand any safety information in the US English publications.

Laser safety information

All IBM Z® and IBM® LinuxONE (LinuxONE) models can use I/O cards such as FICON®, Open Systems Adapter (OSA), Network Express, Integrated Coupling Adapter2.0 SR (ICA SR2.0), zHyperLink Express, or other I/O features which are fiber optic based and utilize lasers (short wavelength or long wavelength lasers).

Laser compliance

All lasers are certified in the US to conform to the requirements of DHHS 21 CFR Subchapter J for Class 1 or Class 1M laser products. Outside the US, they are certified to be in compliance with IEC 60825 as a Class 1 or Class 1M laser product. Consult the label on each part for laser certification numbers and approval information.

Laser Notice: U.S. FDA CDRH NOTICE if low power lasers are utilized, integrated, or offered with end product systems as applicable. Complies with 21 CFR 1040.10 and 1040.11 except for conformance with IEC 60825-1 Ed. 3., as described in Laser Notice No. 56, dated May 8, 2019.

CAUTION: Data processing environments can contain equipment transmitting on system links with laser modules that operate at greater than Class 1 power levels. For this reason, never look into the end of an optical fiber cable or open receptacle. (C027)

CAUTION: This product contains a Class 1M laser. Do not view directly with optical instruments. (C028)

CAUTION
LASER
1M

IEC 1068/14

About this publication

This publication contains information about the Appliance Control Center for IBM Z & IBM LinuxONE.

Unless otherwise stated, throughout this document "9175" refers to the IBM z17® (Model ME1) or IBM LinuxONE Emperor 5 (Model ML1).

Figures included in this document illustrate concepts and are not necessarily accurate in content, appearance, or specific behavior.

Related publications

Publications that you will find helpful and that you should use along with this publication are in the following list. The following publications are available on **IBM Documentation**. Go to https://www.ibm.com/docs/en/systems-hardware, select **IBM Z** or **IBM LinuxONE**, then select your configuration, and click **Library Overview** on the navigation bar.

- Spyre Support Appliance for IBM Z and LinuxONE User's Guide, GC28-7072
- Secure Service Container (SSC) User's Guide, SC28-7062
- IBM Dynamic Partition Manager (DPM) Guide

Related HMC and SE console information

Hardware Management Console (HMC) and Support Element (SE) information can be found on the console help system.

Accessibility features

Accessibility features help users who have physical disabilities such as restricted mobility or limited vision use software products successfully. The accessibility features can help users do the following tasks:

- Run assistive technology such as screen readers and screen magnifier software.
- Operate specific or equivalent features by using the keyboard.
- Customize display attributes such as color, contrast, and font size.

Consult assistive technologies

Assistive technology products, such as screen readers, function with the user interfaces found in this product. Consult the product information for the specific assistive technology product that is used to access our product information.

Keyboard navigation

This product uses standard Microsoft Windows navigation keys.

IBM and accessibility

See http://www.ibm.com/able for more information about the commitment that IBM has to accessibility.

How to provide feedback to IBM

We welcome any feedback that you have, including comments on the clarity, accuracy, or completeness of the information.

For additional information use the following link that corresponds to your configuration:

Configuration	Link
IBM z17® Model ME1	https://www.ibm.com/docs/en/systems-hardware/zsystems/9175-ME1?topic=how-send-feedback
IBM LinuxONE Emperor 5 Model ML1	https://www.ibm.com/docs/en/systems-hardware/linuxone/9175-ML1?topic=how-send-feedback

Summary of changes

Summary of changes for the *IBM Appliance Control Center for IBM Z and LinuxONE User's Guide*, GC28-7073.

Table 1. Summary of changes

Release level	Changes in level
November 2025	This revision contains editorial changes and the following technical changes: • Login to UI task has been added to "Summary of Tasks and Roles on ACC" on page 5 • Updated the APIs in "Updating Appliance by using an Image" on page 58. The APIs use HTTPS instead of HTTP. • The curl commands are updated with -k parameter so that they run in insecure mode.

Chapter 1. Introduction to Appliance Control Center for IBM Z & IBM LinuxONE

The Appliance Control Center for IBM Z & IBM LinuxONE is an application that helps you securely deploy software appliances on an IBM Z and IBM LinuxONE (LinuxONE) servers. It connects to the HMC to manage next generation Appliance Control Center for IBM Z & IBM LinuxONE and Spyre Support Appliance for IBM Z & IBM LinuxONE.

ACC allows users to upload Secure Service Container (SSC) based appliances and updates. After upload, it installs these appliances and their updates. Appliance Control Center for IBM Z & IBM LinuxONE simplifies installation, control, and maintenance of these appliances.

The topics that are covered in this section are:

- "Solution overview" on page 1
- "Modes of Operation" on page 1
- "Connectivity considerations" on page 3
- "Limitations" on page 3
- "Planning for Appliance Control Center for IBM Z & IBM LinuxONE" on page 3
- "Securing Appliance and Appliance Control Center" on page 6

Solution overview

The Appliance Control Center for IBM Z & IBM LinuxONE (ACC) runs on an IBM Z or LinuxONE machine. It connects to the HMC in order to manage appliances on different IBM Z or LinuxONE machines.

The administrator of ACC (also called ACC-admin or simply admin) installs ACC as an LPAR on the IBM Z or LinuxONE machine. The users of ACC use the control node to communicate with ACC. The control node can be a laptop.

The ACC manages appliances on the ACC machine, or on other IBM Z or LinuxONE machines, called appliance machines. The ACC machine and the appliance machines are managed by the same Hardware Management Console (HMC). The appliance machines are optional, and ACC can be managed solely on the ACC machine. Multiple appliance machines are supported.

ACC users download appliance images from IBM Fix Central and upload them to ACC from the control node.

ACC can operate in two modes: default and standalone. For more information, see "Modes of Operation" on page 1.

Modes of Operation

This topic outlines the modes of operation of ACC.

Default mode

In the default mode, ACC has a network connectivity to HMC. This mode provides the maximum convenience and is suitable for automation.

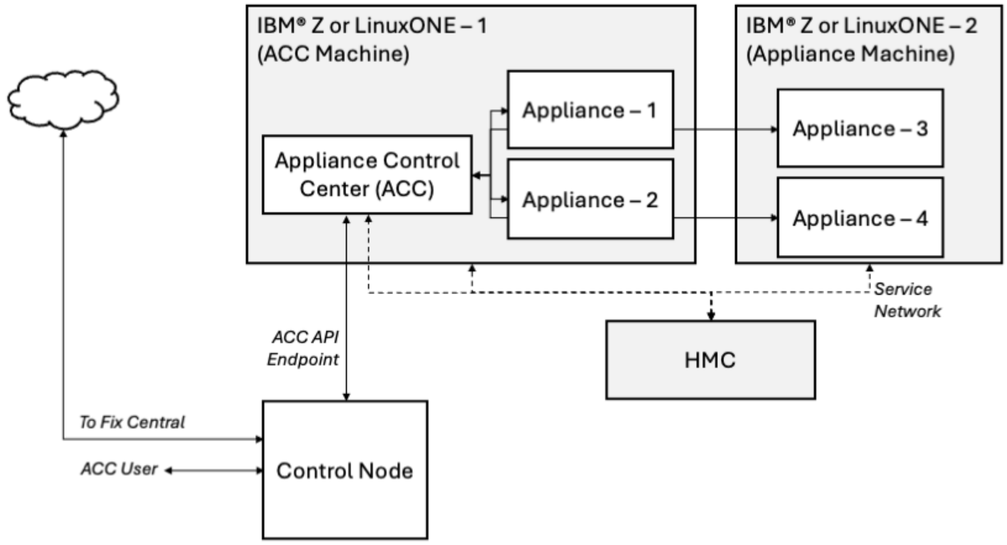


Figure 1. Architecture diagram of ACC in default mode

Stand-alone mode

In stand-alone mode, ACC does not have network connectivity to the HMC. As a result, certain actions on ACC must be preceded by manual steps that are performed on the HMC. This mode is suitable for environments where security, compliance, or organizational processes prevent ACC from communicating directly with the HMC.

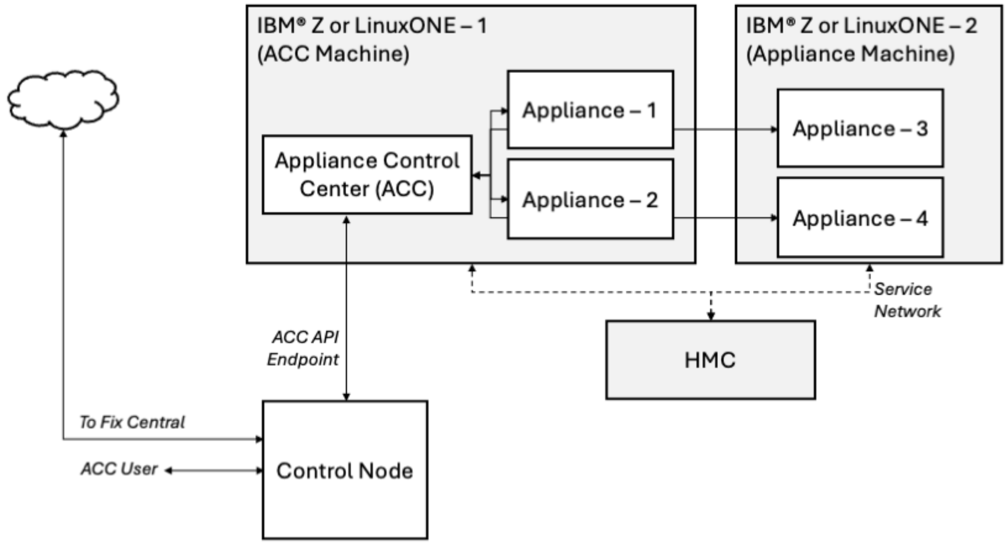


Figure 2. Architecture diagram of ACC in stand-alone mode

Connectivity considerations

To help ensure reliable and efficient data transfer, use the ACC in default mode and place it near the Hardware Management Console (HMC) and the associated appliance machines.

ACC ACC transfers appliance image files to the HMC, which then distributes them to the appliance machines. ACC also retrieves data from the appliances. If the HMC and appliances are not located near the ACC, operations may experience increased latency or timeouts.

If the HMC manages appliances across geographically dispersed locations, consider deploying multiple ACC instances, each serving a specific region. This approach helps maintain performance and reduces the risk of transfer delays or failures.

Limitations

The Appliance Control Center for IBM Z & IBM LinuxONE currently supports only IPv4 networks and does not support a separate data disk. It cannot directly pull updates from Fix Central, and supports only one HMC connection at a time. While the ACC UI has limited functions, all features are accessible through REST APIs.

The ACC has the following limitations:

- ACC supports only IPv4 traffic.
  - ACC administrators cannot assign an IPv6 network to the ACC appliance.
  - ACC cannot install or manage appliances that use IPv6. All appliances must use IPv4.
- ACC does not support a separate data disk.
  - We recommend regularly backing up the ACC disk.
- ACC cannot directly pull updates from Fix Central.
  - As a workaround, you must manually download images and fixes from Fix Central to the control node, then upload them to ACC.
- ACC supports only one HMC connection at a time.
  - If the connected HMC is shut down, for example, for maintenance, ACC cannot perform actions that require HMC access.
- The ACC UI supports limited features.
  - All features are available through REST APIs.
- Only owner can login using the user interface.
- ACC does not support communication with an HMC that enforces multi-factor authentication (MFA). In such cases, standalone mode ACC must be used.

Planning for Appliance Control Center for IBM Z & IBM LinuxONE

The topic outlines the roles and responsibilities within the ACC, focusing on the ACC-admin and appliance owner roles. It details the setup requirements for HMC connectivity, including necessary authorizations and configurations. It also provides a summary of tasks that are performed by ACC, highlighting the interaction between ACC-admin and appliance owner. It includes a planning example for managing resources and appliances by using ACC.

ACC Admin

The ACC administrator manages the ACC machine or CPC and the appliance machines or CPCs. The admin installs ACC on the ACC machine and has login credentials to the HMC that handles the IBM Z machines. ACC communicates with HMC using these credentials. Currently, ACC supports only a single admin.

Some commands require the ACC administrator to be familiar with HMC operations. For detailed instructions, see the HMC Operations Guide.

Appliance Owner

An appliance owner (or simply, owner) is responsible for managing appliances on the machines. Owners can upload, install, activate, deactivate, and delete appliances within ACC. To search for and download appliances from Fix Central, each owner must have valid Fix Central credentials. ACC supports multiple appliance owners.

ACC uses the ACC-admin’s HMC credentials to perform appliance actions such as activating and deactivating. The admin creates multiple owners and assigns each owner a set of resources (IFLs or CPs, memory, LPARs/partitions). Each owner utilizes the assigned resources for their appliances.

Setting up HMC Connectivity

When configuring ACC in the default mode, the admin must ensure that the HMC connectivity is set up properly for ACC to work.

- The web Services API must be enabled on the HMC.
- Port 6794 (TCP) on the firewall between ACC and HMC must be opened.
- The ACC-admin should have a user ID on the HMC and be authorized to perform certain tasks. For more information see “Summary of Tasks and Roles on ACC” on page 5

Machine Authorizations

In default mode, the ACC-admin's user ID on the HMC must be authorized to access the ACC machine and appliance machine. This allows ACC to send requests to the HMC to access these machines for installing, activating, and deactivating appliances on behalf of the appliance owners. This requires the CPC object permissions to be provided to the ACC-admin user ID on the HMC

LPARs or Partitions Authorizations

The ACC-admin's user ID on the HMC must be authorized to access the SSC-type LPARs and partitions managed by ACC. For DPM mode machines, the ACC-admin's user ID should also be authorized for setting up the adapters. ACC will access these SSC-type LPARs or partitions for installing, starting, and stopping appliances.

In the current release, ACC does not create new SSC-type LPARs/partitions; it assumes they are already created. ACC updates activation profiles and runs certain commands listed below.

Authorizations for Classic or Ensemble Mode Machines

The following actions must be authorized for the ACC-admin's HMC user ID for the CPC and ACC or SSA LPARs:

- Customize/Delete Activation Profiles
- Activate
- Deactivate
- Load

Authorizations for DPM Mode Machines

The following actions must be authorized for the ACC-admin's HMC user ID for partitions:

- Partition Details
- Update Partition
- Activate Partition
- Deactivate Partition

Summary of Tasks and Roles on ACC

The following table summarizes the tasks that are performed by ACC:

Task	Role	Mode
Availability of HMC actions using REST APIs	Admin	Default
Authorization to carry out actions against HMC	Admin	Default
Install ACC	Admin	Default and standalone
Create appliance owners in ACC	Admin	Default and standalone
Delete appliance owners in ACC	Admin	Default and standalone
Periodically insert HMC credentials in ACC	Admin	Default
Insert CPC information in ACC	Admin	Standalone
Assign LPARs/partitions to the owners	Admin	Standalone
Enabling and disabling MFA in ACC	Admin	Default and standalone
Inserting CA signed certificates in ACC	Admin	Default and standalone
Download ACC logs	Admin	Default and standalone
Upgrade ACC	Admin	Default and standalone
Assign resources (IFLs/CPs, memory, disks, and so on) to owners	Admin	Default and standalone
Displaying a list of appliances in a catalog	Owner	Default and standalone
Upload appliance images to ACC	Owner	Default and standalone
Install appliances by using uploaded appliance images to ACC	Owner	Default and standalone
Provide the appliance's username and password	Owner	Default and standalone
Assign resources (IFLs/CPs, memory, disk, and so on) to appliances	Owner	Default
Activate/deactivate appliances	Owner	Default
Install fixes and updates to appliances	Owner	Default and standalone
Delete running appliances	Owner	Default
Gather health information about assigned appliances	Owner	Default and standalone
Download appliance logs	Owner	Default and standalone
Login to UI	Owner	Default and standalone

Planning Example

In default mode, if an appliance owner wants to activate multiple appliances, the ACC-admin should take the following actions:

- Install ACC on the ACC LPAR by following the instructions provided in Chapter 2, "Installing Appliance Control Center for IBM Z & IBM LinuxONE," on page 11
- Create the appliance-owner in ACC and name them `app-owner1`.
- Communicate with `app-owner1` to gather resource requirements (total IFLs, total memory, number of LPARs, and so on).

  - Assume that the owner requires four LPARs on a machine that is named `IBMZ1`, each with 16 IFLs and 1 TB memory.
- Verify that enough resources are available on the `IBMZ1` machine by checking the machine information on the HMC
- Assign a total of 64 IFLs and 4 TB memory to `app-owner1` on `IBMZ1` using ACC.
- Assign the 4 SSC-type LPARs/partitions and resources to the owner. The owner will use these SSC-based LPARs/partitions to run the appliances.

Securing Appliance and Appliance Control Center

This topic outlines the security measures required for the Appliance Control Center for IBM Z & IBM LinuxONE. It details the credentials that are used by Appliance Control Center (ACC) for accessing HMC, ACC, and appliances. It emphasizes the importance of securing these credentials by using wallet applications and provides guidelines for password management and API token generation. It also highlights the necessary authorizations and configurations to help ensure secure communication and operation within the ACC environment.

ACC Disk and Network Security

ACC images are encrypted and deployed by using IBM Secure Service Container (SSC) technology. All network communication with ACC is encrypted to help ensure secure data transmission.

⚠ **Attention:**

The disk used to install and activate ACC is encrypted. If the disk is lost or overwritten, the data cannot be recovered.

To prevent data loss, it is recommended to back up the disk regularly. The ACC administrator (ACC admin) should regularly back up the ACC configuration, as outlined in “Creating an ACC Restore Checkpoint” on page 31.

Password Reset

Each API call to ACC requires a token, except for the password reset API `api/user`. Before generating any token for either admin or owner, ACC helps ensure that the password has been reset. The password must:

- be 15 to 128 characters long.
- Include letters, digits, and special characters "(-_#!@$%&?)".
- Not use the username as part of the password.

Before any API call to ACC, the ACC-admin must change the admin's password. ACC stores the hashes of the passwords, not the passwords themselves.

Multi-Factor Authentication (MFA)

ACC supports two-factor authentication (2FA), which the ACC administrator can enable or disable at any time.

ACC generates a secret key for the users. The users then use this secret key to generate Time-Based One-Time Passwords (TOTP). These passwords or tokens must be used along with ACC credentials to log in.

Enabling or disabling 2FA requires all ACC users including administrators and appliance owners to reset or update their passwords. Consider the impact on operations before enabling or disabling MFA. For more information about configuring MFA, see “Configuring Multi-factor Authentication” on page 25.

Networking infrastructure

ACC communicates over the network with the following components:

- HMC
- Appliances

Attention: It is recommended to use an internal network configuration to reduce the risk of external network attacks.

Users interact with ACC through the control node. To reduce the risk of external network attacks, ACC administrators should use separate networks for different types of communication. For example, the HMC network should be isolated from the appliances network. ACC does not route or mix traffic between these networks.

Monitoring and logging with ACC

To help ensure secure and reliable operation, ACC includes robust monitoring and auditing capabilities. As part of its security framework, ACC maintains logs that capture critical security-related events.

ACC administrators can use the "Dumps" feature, available on the ACC host appliance to download these logs.

The ACC host appliance includes an advanced "event monitoring" feature that enhances system visibility and security. This feature enables ACC administrators to generate dumps in response to critical failures or security events, offering deeper insight into the system’s state.

For auditing purposes, ACC administrators can collect dumps that include comprehensive system information, such as:

- Network activity logs
- Storage logs
- Web server and application server logs
- User activity logs

By using these monitoring and audit features, organizations can improve the reliability, integrity, and security of their ACC and appliance environments.

If issues occur with ACC, the ACC administrator can request and download diagnostic logs. ACC uses Secure Service Container (SSC) APIs to perform this operation.

These logs sometimes referred to as dumps.

CA-signed certificates

To secure communication and establish trust, ACC supports uploading a CA-signed certificate. After installation, the ACC administrator must log in and upload the certificate by using the certificate upload API. REST API clients must validate the certificate before accepting any data from ACC.

Verifying Image Integrity

Appliance Control Center for IBM Z & IBM LinuxONE images are digitally signed to verify their authenticity and confirm their origin from IBM.

About this task

To confirm both the integrity and authenticity of the image, you must verify the digital signature.

Procedure

1. Log in to IBM Fix Central.
2. Download both the installation image and its corresponding signature file.

3. Download the public key from: https://public.dhe.ibm.com/systems/z/appliance_control_center/zacc-pub.pub
4. Run the following command to verify the image by using OpenSSL:

   ```
   openssl dgst -sha256 -verify zacc-pub.pub -signature <signature_file> <image_file>
   ```

5. If the image is valid and untampered, you will see:

   ```
   Verified OK
   ```

   This confirms that the downloaded ACC image has not been altered and is indeed from IBM.

Credentials used by Appliance Control Center

To secure the installation of appliances and Appliance Control Center (ACC), it is essential to understand the credentials used by ACC. Once the purpose of these credentials is understood, the ACC user must use a wallet to safely secure these credentials.

HMC Access

Credential	Purpose
HMC_USERID	User ID to access HMC using REST APIs, provided by the ACC-admin.
HMC_USERPASS	Password to access HMC using REST APIs, provided by the ACC-admin.

ACC sends all commands to the HMC using these credentials. ACC will only have a single HMC access credential. These credentials are added to ACC by the ACC-admin by running the API `config/hmcconfig` on the control node. After a successful call to this API, these credentials are stored in the encrypted memory of ACC and are auto-deleted after 24 hours. Therefore, the ACC-admin should run the `config/hmcconfig` API every 24 hours to refresh the HMC access credentials. If ACC is restarted or migrated, these credentials must be reinserted.

Securing HMC Access

Follow general security guidelines and employ the following steps for hardened security:

- The HMC-admin should create a separate user for ACC with the user ID `HMC_USERID`.
- The HMC-admin must limit the actions of `HMC_USERID` to certain authorized actions.
- The ACC-admin must not leave `HMC_USERID` and `HMC_USERPASS` in scripts and Ansible® playbooks.
- The ACC-admin should use a wallet application to secure these credentials.
- The wallet application must exist on a control node that is administered by the ACC-admin only.

Limiting ACC-admin user permissions on HMC

ACC uses the ACC-admin user's HMC credentials to communicate with the HMC. It is recommended that the HMC administrator create these credentials with restricted permissions.

The HMC administrator can apply the following limitations:

- Allow the ACC-admin user to access only the ACC system and appliance systems
- Restrict access to a specific set of LPARs (logical partitions).
- Prevent the ACC-admin user from creating new LPARs.
- Allow the ACC-admin user to update activation profiles only for the selected LPARs.
- Allow the ACC-admin user to activate or deactivate only the selected LPARs.

Note: These restrictions help enforce the principle of least privilege and reduce the risk of unauthorized changes to the HMC environment.

ACC Access

Credential	Purpose
ACC_ADMINUSERID	ACC-admin's user ID to access ACC.
ACC_ADMINUSERPASS	ACC-admin's password to access ACC.
ACC_OWNER1_USERID	User ID of the appliance-owner.
ACC_OWNER1_USERPASS	Password of the appliance-owner.

These credentials are used by the admin or the owner to create an ACC token by using the `user/token` API call. With this token, ACC users can perform their actions. To generate a token, ACC compares the hash of the password in the API call with the stored hash value. The token is valid for 15 minutes after which the user must regenerate the token by using the `user/token` API call. Each user (ACC-admin and appliance-owners) must have separate control nodes and may install a wallet application on their control nodes to store these credentials. Since these credentials are stored as hash values, they can still be used to access ACC if ACC is restarted or migrated.

Appliance Access

Credential	Purpose
APP_USERID	User ID to access the appliance.
APP_USERPASS	Password to access the appliance.

ACC stores the access credentials of the appliances that it manages. Each appliance has an API that requires tokens that are generated by using these credentials. While installing the appliances, the appliance-owner must provide these credentials. ACC stores these credentials on its internal database on an encrypted disk. A restart or migration of ACC will retain these credentials, allowing ACC to manage appliances. It is recommended that appliance-owners use a wallet application to provide these credentials during appliance installation.

Chapter 2. Installing Appliance Control Center for IBM Z & IBM LinuxONE

The topics in this section provide the prerequisites for setting up the Appliance Control Center for IBM Z & IBM LinuxONE and the step by step instructions to configure Appliance Control Center for IBM Z & IBM LinuxONE.

Appliance Control Center (ACC) can be installed using either the provided Ansible playbooks or the Hardware Management Console (HMC) user interface (UI). IBM recommends using the Ansible playbooks for streamlined configuration and automated deployment of ACC.

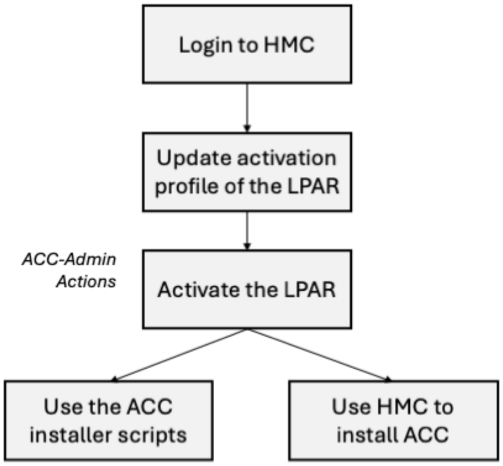


Figure 3. ACC installation flowchart

You should access the HMC user interface and set up the ACC SSC-type LPAR. Then, you can either run the provided installation script or use the UI to install ACC. The topics that are covered in this section are:

- “Checklist” on page 11
- “Downloading Appliance Control Center from Fix Central” on page 12
- “Installing Appliance Control Center for IBM Z & IBM LinuxONE using Ansible playbook” on page 13
- “Installing Appliance Control Center for IBM Z & IBM LinuxONE using HMC” on page 14

Checklist

Use the checklist to review system requirements and confirm that the configuration meets the minimum system requirements.

Before installing the Appliance Control Center for IBM Z & IBM LinuxONE, ensure that the following components are available in the system.

Components	Requirement
IBM z17 , IBM LinuxONE Emperor 4 or higher	Required

Components	Requirement
One SSC-type LPAR or Partition on the ACC machine	Required
At least two shared CPs/IFLs on the ACC machine	Required
At least 16 GB memory on the ACC machine	Required
At least 50 GB DASD or FCP disk that is attached to the ACC machine	Required
Networking device on the ACC machine	Required
Access to HMC (or ACC machine Support Element) UI	Required
To operate in default mode, network connectivity from ACC LPAR/partition to HMC	Required
Connection from ACC machine to the appliance machines	Required
Fix Central credentials to download ACC and appliance	Required
Fix Central credentials to install appliance	Optional
Connection to Fix Central from the control node	Optional
Ansible on control node	Optional

Note: Ensure that the disk system used by ACC is not under stress from other concurrently running LPARs or workloads. Do not use disks larger than 80 GB because formatting them might lead to timeouts.

Downloading Appliance Control Center from Fix Central

You can download the IBM Z Appliance Control Center (ACC) from IBM Fix Central.

Before you begin

- You must have an IBMid.
- Know your product details.

Procedure

1. Go to the Fix Central website.
2. Identify the product: IBM Z Appliance Control Center
   a) On the homepage, use the search bar or the **Find product** tab to locate your product.
   b) Type the product name into the search field (e.g., "IBM Z Appliance Control Center") and select it from the list.
   c) Specify the Installed Version and Platform (operating system). You can select **All** for a broader search. Click **Continue**.
3. Browse for fixes.
4. After identifying the appliance images or fixes, select the ones you want to download and click **Continue**.
5. Sign in and select download options.
6. Accept the terms and conditions and click **Download** now.
7. Download and untar the file.
   a) Follow the on-screen instructions to complete the download.

⚠ **CAUTION:** Do not change the filename. Changing the filename will lead to issues during installation.

a) On your control node, untar the file using the command:

```
tar -xvf Appliance_Control_Center_for_IBMZ_and_IBMLinuxOne_v1.2.6.tar
```

Verify the filename before running the command.

Results

The following files will be untarred on the control node:

- Image with production key - `Appliance_Control_Center_for_IBMZ_and_IBMLinuxOne_v1.2.6.image.img.gz`
- Garasign signatures
- Release Notes with restrictions listed - `Release_Notes_ACC_v1.2.6.txt`
- CMD line tool
- License files
- User's Guide

For installation, use the "img.gz" appliance image file.

Installing Appliance Control Center for IBM Z & IBM LinuxONE using Ansible playbook

You can install Appliance Control Center for IBM Z & IBM LinuxONE by using the Ansible playbook. Download the installation scripts, update the variables, and run the script to install Appliance Control Center for IBM Z & IBM LinuxONE.

Before you begin

- Download and save the repository and scripts that are available on GitHub at https://github.com/IBM/z_ansible_collections_samples/tree/main/z_appliance_control_center on the control node.
- Ensure LPAR activation profile is created and updated with correct network settings
- Ensure that you have the SSC network and disk information.
- Ensure that Ansible is already installed on the control node.
- Verify the integrity of the image by following the instructions provided in "Verifying Image Integrity" on page 7.
- Ensure that ACC is configured in default mode before proceeding. The procedure described below depends on ACC's ability to communicate directly with the HMC. If ACC cannot send commands to the HMC, it is operating in standalone mode, and the appropriate Ansible playbooks must be used instead.

Procedure

1. Export your HMC credentials by running the command:

   ```
   export HMC_USER=<enter_HMC_username>
   export HMC_PASSWORD=<enter_HMC_password>
   ```

2. Modify the variables in `appliance_deploy_default_ansible/acc_env_vars.yaml` to match your environment

   - Use the .img.gz file which was downloaded in "Downloading Appliance Control Center from Fix Central" on page 12 for the "IMAGE_PATH" variable.

3. To install ACC, run the following playbook:

   `ansible-playbook ./appliance_deploy_default_ansible/00_acc_install.yaml`

   Note: This step will initiate the installation process on the SSC type LPAR on the ACC machine. This process may take up to 15 mins.

Results

The Appliance Control Center for IBM Z & IBM LinuxONE is installed.

Installing Appliance Control Center for IBM Z & IBM LinuxONE using HMC

You can install the Appliance Control Center for IBM Z & IBM LinuxONE by using the HMC. Use the ACC-admin account to install Appliance Control Center for IBM Z & IBM LinuxONE.

Before you begin

- Download the Appliance Control Center for IBM Z & IBM LinuxONE installation image to the control node.
- Verify the integrity of the image by following the instructions provided in "Verifying Image Integrity" on page 7.
- Ensure that you meet all the requirements that are defined in the checklist.
- If the target LPAR is not available, it must be created before proceeding with the installation of IBM Z Appliance Control Center (ACC). The method for creating the LPAR depends on the system mode:
  - For non-DPM (standard or classic mode) systems: Use the IODF (Input/Output Definition File) to define and activate the LPAR.
  - For DPM (Dynamic Partition Manager) mode systems: Use the Create Partition function available in the HMC interface.

  Once the LPAR is created and active, you may proceed with the ACC installation process.

For a more detailed procedure, see the Secure Service Container (SSC) User's Guide available on IBM Documentation.

Procedure

1. Login to HMC
2. Select either IBM Z or LinuxONE as the ACC machine.
3. Select the LPAR to install ACC.
4. Update the activation profile of the LPAR:
   a) **General tab**: Set ModeMode to SSC.
   b) **Processor tab**: Select **Not dedicated integrated facility for Linux** and set **Number of processors** - Initial to 2.
   c) **Storage tab**: Assign 16 GB storage.
   d) **SSC tab**: Select **Secure Service Container Installer** in **Boot Selection**,
   e) Provide values for the default master user ID, password and host name.
   f) From the Select Action list in the Network Adapters table, click **Add/Edit Network Adapters** to define a network connection. For each type of network connection in the Secure Service Container environment, supply the following information:
      i) CHPID
      ii) Port
      iii) VLAN ID
      iv) IP Address type, IP Address, Mask, Prefix and Gateway
      v) **DNS server:** From the **Select Action** list in the DNS Servers table, click **Add/Edit DNS server** to define a primary domain name system (DNS) server. The Add/Edit DNS Entry window is displayed. For NETH devices, please provide the Function ID (FID) instead of the CHPID and Port.

g) Save the activation profile.

5. Activate the LPAR

   The ACC appliance credentials are used as the default credentials for ACC. After installation, you can change these credentials at any time without affecting the ACC configuration or functionality.

   Once the LPAR is activated, you can install ACC using either the ACC installer scripts or the HMC.

6. Install the Appliance Control Center for IBM Z & IBM LinuxONE

   a) Open a browser on the control node and enter the LPAR's IP address that was created in Step "4.d" on page 14

   b) Provide the ACC appliance username and password.

   c) Select the installation mode.

   d) Select the disk where the Appliance Control Center for IBM Z & IBM LinuxONE is installed.

   e) Upload the ACC appliance image.

   f) Click **Install**.

What to do next

Setup the Appliance Control Center for IBM Z & IBM LinuxONE by following the instructions provided in Chapter 3, "Configuring Appliance Control Center for IBM Z & IBM LinuxONE," on page 17

Chapter 3. Configuring Appliance Control Center for IBM Z & IBM LinuxONE

The topics in this section provide the prerequisites for setting up the Appliance Control Center for IBM Z & IBM LinuxONE and the step by step instructions to configure Appliance Control Center for IBM Z & IBM LinuxONE.

You can configure the Appliance Control Center (ACC) either by running the provided Ansible playbooks or by manually invoking ACC APIs. The preferred method is using the Ansible playbooks; however, if additional detail is needed or certain steps must be repeated, the reader must use the APIs.

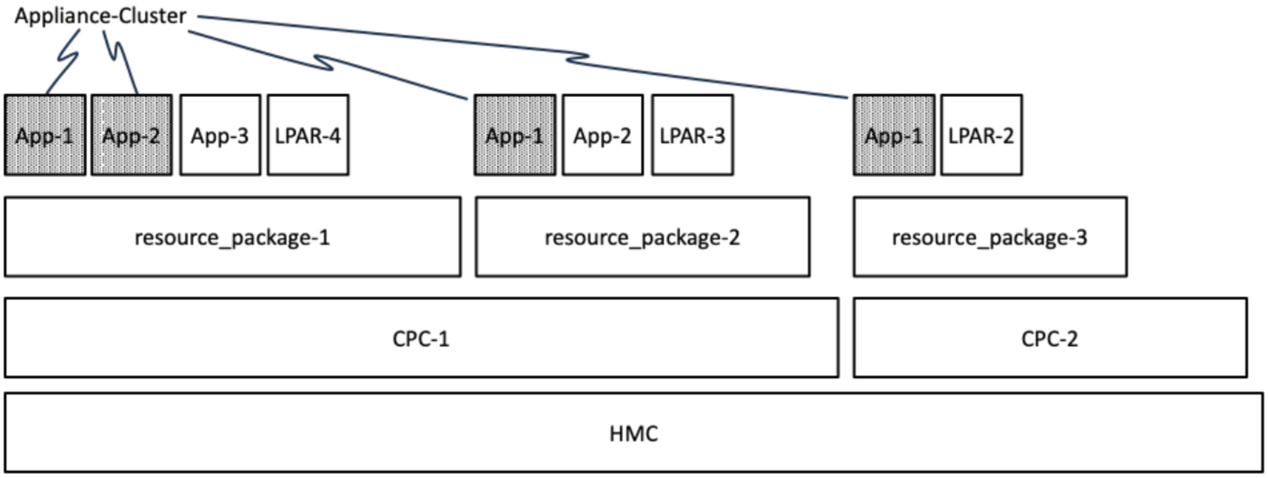
The topics covered in this section are:

- "Configuring Appliance Control Center for IBM Z & IBM LinuxONE using Ansible playbook" on page 18
- "Configuring Appliance Control Center for IBM Z & IBM LinuxONE using the APIs" on page 19
- "Getting Logs of Appliance Control Center" on page 29
- "Configuring Multi-factor Authentication" on page 25
- "Using CA-Signed Certificates" on page 28
- "Upgrading Appliance Control Center for IBM Z & IBM LinuxONE" on page 31
- "Creating an ACC Restore Checkpoint" on page 31

Terminologies

Resource package

ACC manages appliances on IBM Z systems by assigning and orchestrating resources between administrators and appliance owners.



The ACC administrator is responsible for defining and packaging resources such as LPARs, networking, and disk configurations. These resources are grouped into resource packages, each of which includes:

- LPARs
- Networking and disk configurations
- Total cores and memory allocation

Each resource package is assigned to a single appliance owner, who brings their own appliance images and uses the allocated resources to install and run appliances.

ACC is associated with a single HMC, which can manage multiple CPCs including the ACC machine and appliance machines.

Appliance owners can perform operations on:

- Individual LPARs within their resource package.
- Clusters: logical groups of LPARs across one or more resource packages.

For consistency and scalability, you should use cluster-level operations, even for single-appliance tasks.

Understanding Cores (IFLs/GPs) and Memory in Resource Package

The cores and memory assigned in a resource package by the ACC administrator to an appliance owner represent the total number of cores and total memory the owner can use.

For example, if an appliance owner is assigned 16 cores and 4 LPARs it is up to the owner to decide how to distribute these 16 cores among the 4 LPARs when activating them.

In this version of ACC, assigned cores (IFLs or GPs) are considered dedicated resources. For example, if an ACC administrator assigns 16 cores, these are treated as dedicated cores. However, the appliance owner can configure these cores in shared or dedicated mode.

It is not possible to set weights for shared cores in this version of ACC. If required, the ACC administrator should configure LPAR weights using HMC actions.

Configuring Appliance Control Center for IBM Z & IBM LinuxONE using Ansible playbook

You can configure the Appliance Control Center for IBM Z & IBM LinuxONE by using the provided scripts in the Ansible playbook. Download the installation scripts, update the variables in the yaml file, and run the script to install Appliance Control Center for IBM Z & IBM LinuxONE.

Before you begin

- Ensure the Appliance Control Center for IBM Z & IBM LinuxONE is installed and running on the ACC machine.
- Verify that Appliance Control Center for IBM Z & IBM LinuxONE can be reached via the IP address `acc_ip` at port 8081.
- If you want to configure the ACC in default mode, then navigate to "appliance_deploy_default_ansible" directory in the downloaded GitHub repo. For standalone mode ACC, navigate to "appliance_deploy_standalone_ansible" directory.
- To enforce MFA, follow the instructions provided in "Enabling Two-Factor Authentication using Ansible Playbook" on page 26.

Procedure

1. Export your HMC credentials by running the command:

   ```
   export HMC_USER=<enter_HMC_username>
   export HMC_PASSWORD=<enter_HMC_password>
   ```

2. Update the variables present in `admin_vars.yaml` file.

   - `acc_ip` - ACC IP address.
   - `z_machine_lpar` - is the LPAR on which the appliance is installed.
   - Other relevant variables.

   For information on enabling MFA, refer to "Enabling Two-Factor Authentication using Ansible Playbook" on page 26.

3. Run the following Ansible playbook:

```
ansible-playbook 01_admin_actions.yaml
```

4. Adding resource package

   A resource package is a set of resources (like number of IFLs and memory) and LPARs/partitions on a single CPC that is assigned to user. You can think of it as a group of servers.

   Update the `admin_vars.yaml` file with details of both LPARs and then run the following scripts:

   Note: The ACC administrator must exercise caution when assigning resources to appliance owners. Assigning an incorrect resource (for example, adding the wrong LPAR in the REST API) requires deleting and recreating the resource package. For more information on deleting a resource package, see "Deleting resource package" on page 95

   - If you are assigning a single LPAR to the appliance-owner, then run the script:

     ```
     ansible-playbook 02a_assign_1_lpar.yaml
     ```

   - If you are assigning *two* LPARs to the appliance-owner, then run the script:

     ```
     ansible-playbook 02b_assign_2_lpar.yaml
     ```

   If Fiber Channel devices are used to set up the appliances, the ACC administrator must set the `if-fcp` parameter to `true` in the Ansible playbook.

   When `if-fcp` is set to `true`, ACC expects the following additional parameters:

   - `wwpn` - World Wide Port Name. The wwpn of the SCSI disk should be in a 16-digit hexadecimal format. The 0x prefix is not required. For example, 11031568195745bc.
   - `lun` - Logical Unit Number of the storage

   For allocating networking devices to the LPARs, the ACC administrator can either use functional IDs (FIDs) of NETH devices, or use `chpid` and `port`.

5. Reinsert HMC Credentials.

   Note: In default mode, the ansible playbook `01_admin_actions.yaml` will insert the HMC credentials in ACC. These credentials will expire after 24 hours. Therefore, the ACC-administrator must re-enter these credentials into ACC.

   One way of doing that is that ACC adminstrator has to perform the following steps once every day:

   a) Export your HMC and ACC credentials by running the comamnd on your control node:

   ```
   bash export HMC_USER=<enter_HMC_username>
   export HMC_PASSWORD=<enter_HMC_password>
   export ACC_ADMIN_USER=<ACC_admin_username>
   export ACC_ADMIN_PASSWORD=<ACC_admin_password>
   ```

   b) Navigate to `other_usecases_ansible` directory.

   c) Run the following ansible playbook:

   ```
   bash ansible-playbook 09_insert_hmc_creds.yaml
   ```

Configuring Appliance Control Center for IBM Z & IBM LinuxONE using the APIs

You can configure Appliance Control Center for IBM Z & IBM LinuxONE by using the APIs. You need an ACC admin account to configure the ACC appliance.

Before you begin

- Verify that you meet all the requirements that are defined in the checklist.
- Verify that you have admin account credentials

- Verify that Appliance Control Center for IBM Z & IBM LinuxONE is installed and running.
- Verify that ACC can be reached via the IP address `acc_ip` at port 8081.

Procedure

1. Set the following environment variables on the admin's control node:

Variable	Description
`ACC_IP`	IP of ACC
`ACC_PORT`	Port of ACC (8081)
`ACC_ADMINUSERID`	The admin user ID used to access ACC; same as the SSC LPAR user ID before installation.
`ACC_ADMINDEFPASS`	The admin password used to access ACC is the same as the SSC LPAR password before installation.
`ACC_ADMINUSERPASS`	Admin's password to access ACC
`HMC_IP`	IP of the HMC that can be reached by ACC in default mode
`HMC_USERID`	User name to access the HMC by ACC in default mode
`HMC_USERPASS`	Password to access the HMC by ACC in default mode
`CPC`	The CPC used in examples
`ACC_OWNER1_USERID`	User ID of the owner
`ACC_OWNER1_EMAIL`	Email of the owner
`ACC_OWNER1_ASSIGNEDPASS`	Default password of the owner, assigned by the admin.
`MACHINE1`	Name of the machine/CPC on the HMC
`RESOURCE_PKG`	Name of the resource package (server group)
`IFLS`	Total number of assigned IFLs to the RESOURCE_PKG
`GPS`	Total number of assigned GPs to the RESOURCE_PKG
`MEMORY`	Total number of assigned MiB of memory to the RESOURCE_PKG

2. First, the ACC administrator must initialize the ACC. During initialization, the administrator must define:

   - Whether to enable Multi-Factor Authentication (MFA) for ACC.
   - Whether ACC should operate in default mode or standalone mode.

   Important: This initialization call can be made only once. If the request parameters are incorrect, the ACC administrator must reinstall ACC.

   Example initialization call -

   To disable MFA and enable default mode, use the following `curl` command:

```
curl -k -X 'POST' \
  "https://$ACC_IP:$ACC_PORT/api/init" \
  -H 'accept: application/json' \
  -H 'Content-Type: application/json' \
  -d '{
      "mfa_enable": false,
      "hmc_managed": true,
      "credentials": {
         "username": "'$ACC_ADMINUSERID'",
         "password": "'$ACC_ADMINDEFPASS'"
      }
   }'
```

In this example:

- `mfa_enable` is set to `false` to disable MFA.

  Note: MFA can be enabled or disabled using a REST API call as outlined in "Configuring Multi-factor Authentication" on page 25.

- `hmc_managed` is set to `true` to enable default mode.

  Important: This setting cannot be changed later.

- The `credentials` field verifies that the ACC administrator performing the installation is authorized to initialize ACC.

Note: The default username and password are the same as those configured by the ACC administrator in the activation profile of the SSC logical partition (LPAR) of ACC.

3. Before creating an ACC token for the first time, each user must change their password. As the ACC admin, use the default username `ACC_ADMINUSERID` and default password `ACC_ADMINDEFPASS` to change the password through the Password Change API.

   The default credentials for ACC-admin are the same as the SSC LPAR username and password configured by the ACC admin in the LPAR activation profile.

   ```
   curl -k -X 'PUT' \
   "https://$ACC_IP:$ACC_PORT/api/user" \
   -H 'accept: application/json' \
   -H 'Content-Type: application/json' \
   -d '{
     "username": "'$ACC_ADMINUSERID'",
     "old_password": "'$ACC_ADMINDEFPASS'",
     "new_password": "'$ACC_ADMINUSERPASS'"
   }'
   ```

   If MFA is enabled, then you need to include additional parameter `otp`.

   Password requirements:

   - Length: 15-128 characters
   - Valid characters: letters, digits, special characters (`-_#!@$%&?`)
   - Must have at least one lower case, one upper case, one digit, one special character
   - Must not include the username

   Note: Changing the password is not required every time the ACC admin logs in. You can skip this step and perform it only when necessary. Additionally, note that if MFA is enabled, then the password update API will send back a new secret key. For more information, see "Configuring Multi-factor Authentication" on page 25.

4. Login to the ACC using admin credentials and obtain a token to perform authentication and authorization using `login API`.

   ```
   response=$(curl -k -X 'POST' \
   https://$ACC_IP:$ACC_PORT/api/user/token \
   -H 'accept: application/json' \
   -H 'Content-Type: application/json' \
   -d '{
     "username": "'$ACC_ADMINUSERID'",
     "password": "'$ACC_ADMINUSERPASS'"
   }')
   ADMIN_TOKEN=$(echo $response | jq '.access_token' | tr -d '"')
   ```

   The `$ADMIN_TOKEN` will be used for authentication and authorization in subsequent steps.

5. Add CPC information:

<table>
<tr><th>Mode</th><th>API</th></tr>
<tr><td><b>Default</b></td><td>

Insert the HMC credentials into ACC. Use the token that was generated in “4” on page 21 for authentication.

Note: The caller should use double quotes around the variables, for example "$HMC_USERID". However, if the caller is using special characters in these variables, then care must be taken and these characters must be properly escaped.

```
curl -k -X 'POST' \
  "https://$ACC_IP:$ACC_PORT/api/config/hmcconfig" \
  -H 'accept: */*' \
  -H "Authorization: Bearer $ADMIN_TOKEN" \
  -H 'Content-Type: application/json' \
  -d '{
        "host": "'$HMC_IP'",
        "userid": "'$HMC_USERID'",
        "password": "'$HMC_USERPASS'",
        "verify_cert": false
   }'
```

If the HMC IP and credentials are valid, then the API fetches CPC information from the HMC and stores it in the ACC.

Note:

- Currently, only false is supported for verify_cert parameter.
- HMC credentials expire in 24 hours and must be re-entered after this period.

Verify whether the HMC is reachable from the ACC by using the following API:

```
curl -k -X 'GET' \
"https://$ACC_IP:$ACC_PORT/api/cpcs/hmc-connection" \
-H "Authorization: Bearer $ADMIN_TOKEN" \
-H 'accept: application/json'
```

</td></tr>
<tr><td><b>Standalone</b></td><td>

In standalone mode (when `hmc_managed` is `false`), the ACC administrator must manually add CPC information to ACC. To do this, use the `cpc` API:

```
curl -k -X 'POST' \
  "https://$ACC_IP:$ACC_PORT/api/cpcs" \
  -H 'accept: application/json' \
  -H 'Content-Type: application/json' \
  -H "Authorization: Bearer $ADMIN_TOKEN" \
  -d '{
         "cpcs": [
            {
               "cpc_name": "'$MACHINE1'",
               "ifls": 0,
               "gps": 108,
               "available_storage": 229376,
               "dpm_enabled": false
            }
         ]
       }'
```

This API allows the administrator to add details for one or more CPCs, such as

`ifls`, `gps`, and whether the CPC is `dpm_enabled`. The information is stored in the ACC internal database and used for validation.

</td></tr>
</table>

6. Required: (Default mode) Retrieve information about the available resources (IBM® Z or LinuxONE machines, cores, memory) using the ACC.

   The following API retrieve the list of CPCs that are managed by ACC.

```
curl -k -X 'GET' \
"https://$ACC_IP:$ACC_PORT/api/cpcs" \
-H 'accept: application/json' \
-H "Authorization: Bearer $ADMIN_TOKEN"
```

The output consists of list of CPCs that are managed by ACC.

⚠ **Attention:** If the CPC is removed from the HMC after performing “5” on page 21, the CPC continues to be listed in the ACC. You must verify whether the CPC is active before proceeding with further steps.

The following API retrieves information on a specific CPC that is stored in ACC:

```
curl -k -X 'GET' \
"https://$ACC_IP:$ACC_PORT/api/cpcs/$MACHINE1/resource" \
-H 'accept: application/json' \
-H "Authorization: Bearer $ADMIN_TOKEN"
```

If you encounter any issues in viewing CPC information, then see “Unable to retrieve CPC information” on page 95

7. Create owners by using the `create owner API()`.

Run the following using the admin account.

```
curl -k -X 'POST' \
 "https://$ACC_IP:$ACC_PORT/api/user" \
 -H 'accept: */*' \
 -H "Authorization: Bearer $ADMIN_TOKEN" \
 -H 'Content-Type: application/json' \
 -d '{
 "username": "'$ACC_OWNER1_USERID'",
 "email": "'$ACC_OWNER1_EMAIL'",
 "password": "'$ACC_OWNER1_ASSIGNEDPASS'",
 "role": "owner"
 }'
```

Owner Username Requirements:

- Length: 6-16 characters
- Valid characters: lowercase letters, digits, '-', '_'
- Starts with a lowercase letter
- Ends with a lowercase letter or digit
- The username and email must be unique.

  Note: The owner email must contain the “@” symbol, and the domain portion must include both a name and a top-level domain (TLD), separated by a dot (e.g., example.com).

Owner Password Requirements:

- Length: 6-128 characters
- Valid characters: letters, digits, special characters (`-_#!@$%&?`)
- Must not include the username

8. Assign resources to the owners by creating a resource package and assigning it to the owner.

In this example, the admin assigns the resource package *RESOURCE_PKG* to the owner for the machine *MACHINE1* on the HMC, which can be either an ACC or appliance machine. The owner can use **IFLS** , **GPS** general-purpose cores, **MEMORY** MiB memory, and logical partitions *LP20* and *LP53*.

ACC assumes *LP20* and *LP53* are predefined and configured on the HMC. Networking information in interfaces and boot disk information in `boot-info` can be dynamically attached to the logical partitions. Both must be available before using the command.

The **ip** and **gw** parameters in interface denote the IP address and gateway of the appliances once started by the owner.

If both logical partitions run appliances, they must not exceed **IFLS**, **GPS** or **MEMORY** MiB memory combined.

Note: Assigning a **RESOURCE_PKG** does not activate the LPARs. It only indicates that the LPARs belong to a specific owner.

An LPAR might already be active through HMC actions before being assigned to an owner in the **RESOURCE_PKG**. In such cases:

Default ACC mode:
ACC retrieves disk (boot-info) and network (interfaces) details from the HMC. If these values differ from those in the **RESOURCE_PKG**, ACC automatically updates the **RESOURCE_PKG** values.

Standalone ACC mode:
ACC updates only the disk information if it differs from the retrieved value.

Recommendation
Administrators should enter all values, even if unused, to avoid confusion. For example, if the owner is not allowed to use `gps`, explicitly set `gps` to `0`.

For DPM-mode machines, the administrator must ensure that partition names are not changed on the HMC by any user. Changing partition names can cause unexpected behavior in ACC.

The admin can assign multiple resource packages to the same owner, but a package can only contain resources from a single machine.

```
curl -k -X 'POST' \
"https://$ACC_IP:$ACC_PORT/api/resource/pkgs" \
-H 'accept: */*' \
-H "Authorization: Bearer $ADMIN_TOKEN" \
-H 'Content-Type: application/json' \
-d '{
  "owner": "'$ACC_OWNER1_USERID'",
  "name": "'$RESOURCE_PKG'",
  "ifls": $IFLS,
  "gps": $GPS,
  "memory": $MEMORY,
  "lpars": [
    {
      "name": "LP20",
      "interfaces": [
        {
          "name": "interface1",
          "chpid": "03",
          "port": 0,
          "vlan_id": 300,
          "prefix": 22,
          "ip": "192.xx.xx.185",
          "gw": "192.x.xx.1"
        }
      ],
      "boot-info": {
        "disk-id": "0.0.2910",
        "is-fcp": false
      }
    },
    {
      "name": "LP53",
      "interfaces": [
        {
          "name": "enc700",
          "fid": "0105",
          "prefix": 23,
          "ip": "192.xx.xx.210",
          "gw": "193.xx.xx.5"
        }
      ],
      "boot-info": {
        "disk-id": "0.0.6f48",
        "is-fcp": false
      }
    }
  ],
  "cpc": "'$MACHINE1'"
}'
```

Resource package name requirements:

- Valid characters: letters (a-z, A-Z), digits (0-9), '-', '_'

If Fiber Channel devices are used for setting up the appliances, then the ACC-admin should set the `if-fcp` parameter to `true` in the resource package assignment API.

If `if-fcp` is `true`, then ACC expects additional parameters: `wwpn` (World Wide Port Name) and `lun` (Logical Unit Number of the storage). These parameters should be provided under the `boot-info`.

In the above example, you can see that LP20 is assigned to `chpid` and `port`, whereas the LP53 is assigned an `fid` (NETH device).

Note: The ACC administrator must exercise caution when assigning resources to appliance owners. Assigning an incorrect resource (for example, adding the wrong LPAR in the REST API) requires deleting and recreating the resource package. For more information on deleting a resource package, see "Deleting resource package" on page 95

What to do next

You can continue as an ACC-admin to configure the ACC by following the instructions provided in this topic. Once the configuration is complete, install the appliances by following the instructions provided in Chapter 4, "Installing an Appliance using Appliance Control Center for IBM Z & IBM LinuxONE," on page 33

Configuring Multi-factor Authentication

ACC supports Two-Factor Authentication (2FA) to enhance login security. When 2FA is enabled, in addition to username and password, users must provide a second form of authentication to obtain an API token.

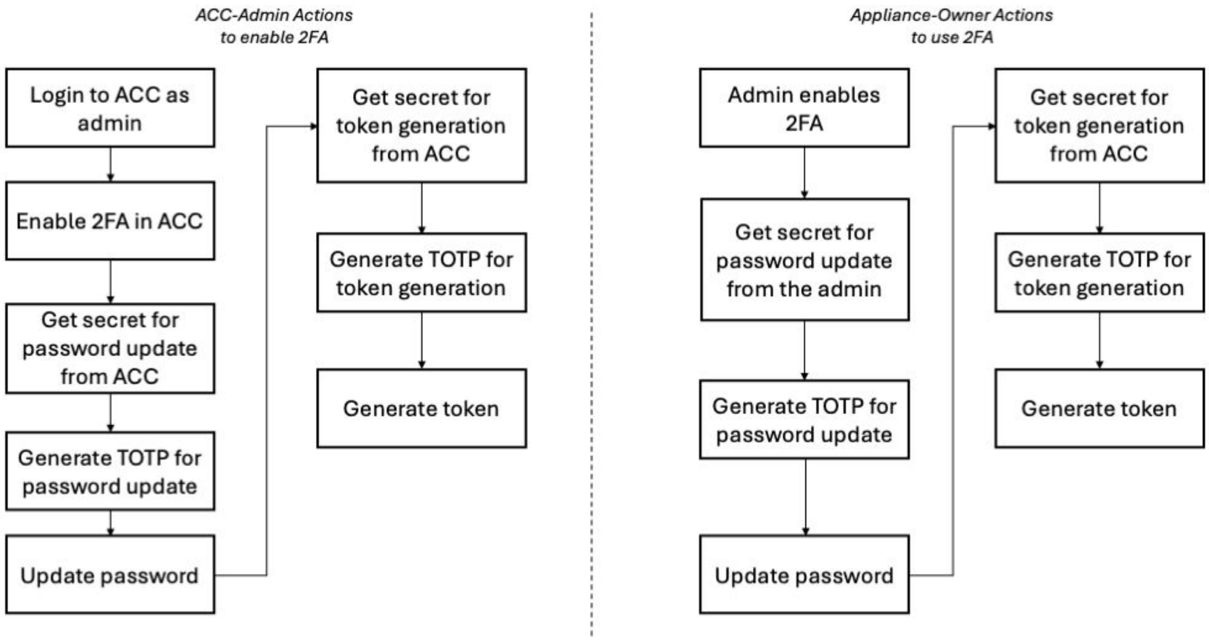
ACC admins can enable or disable 2FA, thereby controlling whether appliance owners must use 2FA to generate their ACC tokens.

ACC uses Time-Based One-Time Passwords (TOTP) defined in RFC 6238. TOTP generates a temporary code based on the current time and a secret key. Each code is valid for 30 seconds.

When a user updates their password and 2FA is enabled, ACC generates a secret key and provides it to the user. The user must add this key to a TOTP generator, which can be an authenticator app on the user's mobile phone. The TOTP generator then generates a new TOTP every 30 seconds.

After 2FA is enabled:

- ACC admin receives a temporary secret key to update admin's password using the TOTP generated by the temporary secret key. Once updated, the password update API returns a new secret key for use in the authenticator app.
- ACC admin can also call a REST API to generate temporary secret keys for appliance owners. These keys are shared with the respective owners, who then use them to update their passwords and configure their own TOTP generators.


ACC-Admin Actions
to enable 2FA
Login to ACC as admin
Enable 2FA in ACC
Get secret for password update from ACC
Generate TOTP for password update
Update password
Get secret for token generation from ACC
Generate TOTP for token generation
Generate token
Appliance-Owner Actions
to use 2FA
Admin enables 2FA
Get secret for password update from the admin
Generate TOTP for password update
Update password
Get secret for token generation from ACC
Generate TOTP for token generation
Generate token

The topics covered in this section are:

- "Enabling Two-Factor Authentication using Ansible Playbook" on page 26
- "Enabling two-factor authentication using APIs" on page 27

Enabling Two-Factor Authentication using Ansible Playbook

The ACC admin can enable two-factor authentication by using the Ansible playbooks.

Before you begin

- Download and save the repository and scripts that are available on GitHub at https://github.com/IBM/z_ansible_collections_samples/tree/main/z_appliance_control_center on the control node.

Procedure

1. Depending on the ACC mode, navigate to the appropriate folder:
   - For default mode ACC, run

     ```
     cd appliance_deploy_default_mfa_ansible
     ```

   - For standalone mode ACC, run:

     ```
     cd appliance_deploy_standalone_mfa_ansible
     ```

2. Configure Variables
   Edit the file `admin_vars.yaml` and adjust the variables as needed.
3. Initialize ACC with MFA
   Run the following playbook:

   ```
   ansible-playbook 01_admin_actions.yaml
   ```

4. Assign LPARs to Appliance Owner
   - To assign single LPAR to appliance owner, run

```
ansible-playbook 02a_assign_1_lpar.yaml
```

- To assign two LPAR to appliance owner, run

```
ansible-playbook 02b_assign_2_lpar.yaml
```

Enabling two-factor authentication using APIs

The ACC admin can enable two-factor authentication by using the API provided in the procedure.

Procedure

To enable 2FA for generating ACC tokens, use the following API:

```
curl -k -X 'PUT' \
"https://$ACC_IP:$ACC_PORT/api/mfa/enable" \
-H 'accept: */*' \
-H 'Content-Type: application/json' \
-d '{
"username": "'$ACC_ADMINUSERID'",
"password": "'$ACC_ADMINUSERPASS'"
}'
```

Generating MFA Secrets for Admins and Appliance Owners

To use Time-Based One-Time Passwords (TOTP) for two-factor authentication (2FA), users must first obtain a secret key and configure it in an authenticator app.

Procedure

1. Generate MFA Secret for ACC Admin

   The ACC admin can generate a secret key for their own authenticator app by using the following API:

   ```
   curl -k -X 'POST' \
     "https://$ACC_IP:$ACC_PORT/api/mfa/secret/admin" \
     -H 'accept: */*' \
     -H 'Content-Type: application/json' \
     -d '{
       "username": "'$ACC_ADMINUSERID'",
       "password": "'$ACC_ADMINUSERPASS'"
   }'
   ```

   This secret key must be added to a TOTP-compatible authenticator application to generate time-based codes.

2. Generate MFA Secret for Appliance Owner

   The ACC admin can also generate a temporary secret key for an appliance owner. This key allows the owner to:

   - Generate a TOTP
   - Update their own secret key

   To generate the secret key for an appliance owner, the ACC admin uses the following API:

   ```
   curl -k -X 'POST' \
     "https://$ACC_IP:$ACC_PORT/api/mfa/secret/owner" \
     -H 'accept: */*' \
     -H "Authorization: Bearer $ADMIN_TOKEN" \
     -H 'Content-Type: application/json' \
     -d '{
       "username": "'$ACC_OWNER1_USERID'"
   }'
   ```

The admin must share this secret key with the appliance owner using a secure messaging platform. The owner then uses this key to configure their authenticator app and complete the password update process.

Disabling two-factor authentication

The ACC admin can disable two-factor authentication by using the API provided in the procedure.

Procedure

To disable 2FA for ACC token generation, the ACC admin can use the following API:

```
curl -k -X 'PUT' \
 "https://$ACC_IP:$ACC_PORT/api/mfa/disable" \
 -H 'accept: */*' \
 -H "Authorization: Bearer $ADMIN_TOKEN" \
 -H 'Content-Type: application/json' \
 -d '{
 "username": "'$ACC_ADMINUSERID'",
 "password": "'$ACC_ADMINUSERPASS'"
}'
```

Using CA-Signed Certificates

An ACC administrator can upload CA-signed certificates to ACC. To do this, the administrator must instruct ACC to generate a Certificate Signing Request (CSR). After obtaining the signed CSR from a Certificate Authority (CA), the administrator uploads it to ACC.

Note: If multiple IP addresses are configured, ACC supports only one certificate at a time

The topics covered in this section are:

- “Enabling two-factor authentication using APIs” on page 27
- “Generating MFA Secrets for Admins and Appliance Owners” on page 27
- “Disabling two-factor authentication” on page 28

Generating CSR in ACC

You can generate a Certificate Signing Request (CSR) by using the API provided in the procedure.

Procedure

The ACC administrator can generate a CSR by sending the following command to ACC:

~~~
curl -k -X 'POST' \
"https://$ACC_IP:$ACC_PORT/api/certificate/csr" \
-H 'accept: */*' \
-H "Authorization: Bearer $ADMIN_TOKEN" \
-H 'Content-Type: application/json' \
-d '{
  "country": "DE",
  "state": "Baden-Württemberg",
  "ip": "0.0.0.0"
}'
```
~~~

If the request is successful, ACC returns a CSR similar to the following:

```
-----BEGIN CERTIFICATE REQUEST-----
...QAwWTELMAkGA1UEB...
-----END CERTIFICATE REQUEST-----
```

The administrator must then send the CSR to a certificate authority (CA) for signing. The CA returns a signed certificate.

Uploading the CA-Signed Certificate to ACC

You can upload the CA-signed certificate by using the API provided in the procedure.

Procedure

After obtaining the signed certificate from "Generating CSR in ACC" on page 28, the ACC administrator can upload it to ACC using the following API:

```
curl -k -X 'POST' \
"https://$ACC_IP:$ACC_PORT/api/certificate/upload" \
-H 'accept: */*' \
-H "Authorization: Bearer $ADMIN_TOKEN" \
-H 'Content-Type: application/octet-stream' \
--data-binary "@path/to/signed_certificate_file"
```

Before uploading, the user must ensure that the certificate is in PEM format. To verify the format, open the certificate file in a text editor and check for the following markers:

```
-----BEGIN CERTIFICATE-----
```

and

```
-----END CERTIFICATE-----
```

If the file appears as binary and lacks these markers, it is likely DER-encoded. In such cases, use the OpenSSL tool to convert it to PEM format before re-uploading.

Example Open SSL command `openssl x509 -inform DER -in certificate.crt -out certificate.pem`

Note:

Files with the `.crt` or `.cer` extension are often interchangeable. The above command works for both.

Once converted to PEM format, retry the certificate upload using the API.

Getting Logs of Appliance Control Center

IBM Z Appliance Control Center (ACC) might encounter errors that require collecting logs for troubleshooting. These logs can be gathered from ACC and shared with IBM for analysis.

ACC is a Secure Service Container (SSC)-based appliance, you can use SSC APIs to collect logs.

You can gather logs by using one of the following method:

- "Gathering logs using Ansible Playbook" on page 29
- "Gathering logs manually" on page 30

Gathering logs using Ansible Playbook

You can download logs by using the Ansible playbook.

Before you begin

- Download the Ansible playbooks that are available on GitHub at https://github.com/IBM/z_ansible_collections_samples/tree/main/z_appliance_control_center .
- Ensure that you have logged as ACC-admin to run the Ansible playbook.

Procedure

1. Download the scripts from the `other_usecases_ansible` directory to your control node.
2. Navigate to the downloaded directory:

```
cd other_usecases_ansible
```

3. Read the README.md file for details on running the scripts and modifying configuration files.
4. Run the Ansible playbook. For example:

```
ansible-playbook 03_pull_ssc_logs.yaml
```

Gathering logs manually

You can gather logs by using the curl commands.

Before you begin

Define the following variables:

Variable	Purpose
ACC_IP	IP address of ACC
ACC_LPAR_USERNAME	Username of the SSC LPAR on the HMC profile (used during ACC install)
ACC_LPAR_PASSWORD	Password of the SSC LPAR on the HMC profile (used during ACC install)
REASON	Reason for collecting ACC logs **Note:** This should not be more than 256 characters.

Procedure

1. Log in to the Appliance

   Use the SSC REST API to generate an authentication token:

```
curl -k -X 'POST' \
  "https://$ACC_IP/api/com.ibm.zaci.system/api-tokens" \
  -H 'accept: application/vnd.ibm.zaci.payload+json' \
  -H 'Content-Type: application/vnd.ibm.zaci.payload+json;version=1.0' \
  -H 'zACI-API: com.ibm.zaci.system/1.0' \
  -d '{
         "user": "'$ACC_LPAR_USERNAME'",
         "password": "'$ACC_LPAR_PASSWORD'"
  }'
```

   If successful, store the returned token in a variable called SSC_TOKEN.

2. Send logging requests to ACC:

   Trigger log preparation by sending an alert:

```
curl -k -X 'POST' \
  "https://$ACC_IP/api/com.ibm.zaci.system/alerts" \
  -H "accept: application/vnd.ibm.zaci.payload+json" \
  -H "Content-Type: application/vnd.ibm.zaci.payload+json;version=1.0" \
  -H "zACI-API: com.ibm.zaci.system/1.0" \
  -H "Authorization: Bearer $SSC_TOKEN" \
  -d '{
         "reason": "'$REASON'",
         "diag_info": "concurrent"
  }'
```

   The response includes a uuid, which you should store in UUID.

3. Check Alert Status

   Monitor the alert status:

```
curl -k -X 'GET' \
  "https://$ACC_IP/api/com.ibm.zaci.system/alerts/$UUID" \
  -H "accept: application/vnd.ibm.zaci.payload+json" \
  -H "zACI-API: com.ibm.zaci.system/1.0" \
  -H "Authorization: Bearer $SSC_TOKEN"
```

When the status is `200` or `202`, ACC has finished gathering logs.

4. Download the Logs

   Download the log file:

```
curl --fail-with-body -k -X 'GET' \
  "https://$ACC_IP/api/com.ibm.zaci.system/alerts/$UUID/diag-info" \
  -H "accept: application/vnd.ibm.zaci.payload+json" \
  -H "zACI-API: com.ibm.zaci.system/1.0" \
  -H "Authorization: Bearer $SSC_TOKEN"
  --output acc.log
```

Note: The downloaded file is an archive with .zip extension.

Upgrading Appliance Control Center for IBM Z & IBM LinuxONE

You can upgrade the Appliance Control Center for IBM Z & IBM LinuxONE to a newer version by downloading the latest version from the Fix Central.

Before you begin

- Download the latest bundle from the Fix Central.
- Download the Ansible playbook scripts.

Procedure

1. Navigate to `other_usecases_ansible`
2. Run

```
ansible-playbook 06_acc_appliance_update.yaml
```

Creating an ACC Restore Checkpoint

You may need to revert IBM Z Appliance Control Center (ACC) to a previous state or configuration. This may be necessary if an upgrade or incorrect input corrupts the ACC configuration.

Before you begin

Variable	Purpose
ACC_IP	IP address of ACC
ACC_LPAR_USERNAME	Username of the SSC LPAR on the HMC profile (used during ACC install)
ACC_LPAR_PASSWORD	Password of the SSC LPAR on the HMC profile (used during ACC install)
REASON	Reason for capturing the ACC configuration

About this task

To prepare for such scenarios, ACC administrators should regularly create and securely store checkpoints of the ACC configuration. You can automate this process by creating a job that collects and stores these checkpoints.

Procedure

1. Log in to the Appliance

Generate an appliance admin token (SSC_TOKEN) using the SSC REST API:

```
curl -k -X 'POST' \
  "https://$ACC_IP/api/com.ibm.zaci.system/api-tokens" \
  -H 'accept: application/vnd.ibm.zaci.payload+json' \
  -H 'Content-Type: application/vnd.ibm.zaci.payload+json;version=1.0' \
  -H 'zACI-API: com.ibm.zaci.system/1.0' \
  -d '{
    "kind": "request",
    "parameters": {
      "user": "admin",
      "password": "P@55word"
    }
  }'
```

Store the returned token in SSC_TOKEN. This token is required for the next steps.

2. Export the Configuration

   Run the following command to export the ACC configuration:

```
curl -k -X 'POST' \
  "https://$ACC_IP/api/com.ibm.zaci.system/appliance-configuration/export" \
  -H "Authorization: Bearer $SSC_TOKEN" \
  -H "zACI-API: com.ibm.zaci.system/1.0" \
  -H "Content-type: application/vnd.ibm.zaci.payload+json;version=1.0" \
  -H "Accept: application/octet-stream" \
  -o "acc-config-$(date +%d-%m-%Y).data" \
  -d '{
    "kind": "request",
    "parameters": {
      "description": "'Create ACC configuration data'"
    }
  }'
```

   This command:

   - Creates a configuration snapshot on the ACC.
   - Downloads the configuration file as `acc-config-$(date +%d-%m-%Y).data`.

   Important: Store this file securely.

3. Restore the Configuration

   To restore ACC to a previous state, import the saved configuration:

```
curl -k -X 'POST' \
  "https://$ACC_IP/api/com.ibm.zaci.system/appliance-configuration/import?apply_now=true" \
  -H "Authorization: Bearer $SSC_TOKEN" \
  -H "zACI-API: com.ibm.zaci.system/1.0" \
  -H "Accept: application/vnd.ibm.zaci.payload+json" \
  -H "Content-Type: application/octet-stream" \
  --data-binary @acc-config-25-09-2025.data
```

   Warning:

   The appliance will restart after the configuration is applied.

   Note: Currently, ACC configuration capture and restore is only supported within the same version of ACC. Cross-version configuration migration is not supported at this time.

Chapter 4. Installing an Appliance using Appliance Control Center for IBM Z & IBM LinuxONE

The topics in this section provide the prerequisites for setting up the Appliance Control Center for IBM Z & IBM LinuxONE and the step by step instructions to configure Appliance Control Center for IBM Z & IBM LinuxONE.

You can install appliances by running the provided Ansible playbooks, manually invoking ACC APIs, using ACC's UI, or by using HMC's UI. The preferred method is using the Ansible playbooks; however, if additional detail is needed or certain steps must be repeated, you must use the APIs.

Before you begin

- Appliance Control Center for IBM Z & IBM LinuxONE is installed and running.
- ACC admin has created the appliance owner account.
- A resource package is assigned to the owner.
- The appliance image is available locally on the control node.
- If the LPAR is not available in the HMC, then it must be created before you can proceed with installing an appliance. This can be achieved by using IODF on non-DPM machines, and using the **Create Parition** function on DPM mode machines.

The topics that are covered in this section are:

- "Install Appliance using Ansible Playbooks" on page 33
- "Installing an Appliance using API commands" on page 34
- "Installing Appliances using ACC UI" on page 37

Install Appliance using Ansible Playbooks

The appliance must be installed by the appliance owner. To complete the installation, perform the following steps:

Before you begin

- Download the Appliance Image from the Fix Central.
- Download and save the repository and scripts that are available on GitHub at https://github.com/IBM/z_ansible_collections_samples/tree/main/z_appliance_control_center on the control node.

Procedure

1. If using default mode ACC, navigate to `ansible_deploy_default_ansible` directory. If using standalone mode ACC, then navigate to the `ansible_deploy_standalone_ansible` directory.

   ```
   cd appliance_deploy_default_ansible
   ```

2. Update configuration files

   - Modify the variables in the `owner_vars.yaml` file.
   - Update the `acc_ip` in the `admin_vars.yaml` file to point to the correct ACC IP address.
   - Update the `image_path` variable in the `owner_vars.yaml` file to specify the location of the downloaded image.

3. Run the following playbook to update owner credentials and upload the appliance image:

   ```
   ansible-playbook 03_owner_action.yaml
   ```

4. Run the following playbook to install the appliance on the target LPAR

   ```
   ansible-playbook 04_install_flow.yaml
   ```

What to do next

To pull logs, perform concurrent updates, upgrade, check health status, or update ACC, use the scripts located in:

```
other_usecases_ansible
```

Installing an Appliance using API commands

This section describes how to upload and install and appliance using Appliance Control Center for IBM Z & IBM LinuxONE.

Before you begin

Note: The process for installing appliances differs between the default and standalone modes of ACC. In default mode, no manual action is required on the HMC. In standalone mode, the LPARs must be set to active in "SSC Installer" mode through the HMC UI and must be accessible using the IP address configured in the LPAR’s network settings.

For a more detailed procedure, see the Secure Service Container (SSC) User's Guide available on IBM Documentation.

Procedure

1. Set Environment Variables

   Define the following environment variables on the control node:

Variable	Purpose
ACC_IP	IP address of the ACC
ACC_PORT	Port number (8081)
ACC_OWNER1_USERID	User ID of the appliance owner (assigned by ACC admin)
ACC_OWNER1_ASSIGNEDPASS	Initial password assigned by ACC admin
ACC_OWNER1_USERPASS	New password set by the owner
LPAR	Logical partition name for appliance installation
CORES	Number of cores (IFLs or GPs) for the LPAR
MEMORY	Amount of memory (in MiB) for the LPAR
APP_USERID	User ID to access the appliance
APP_USERPASS	Password to access the appliance
RP1	Name of the resource package that is assigned to the owner.
IMG_LOC	Location of the appliance image on control node

   You can store these in a shell script and run `source <script>` to load them into your environment.

2. Change Owner Password (First-Time Only)

   Use the password change API to update the assigned password:

   ```
   curl -k -X 'PUT' "https://$ACC_IP:$ACC_PORT/api/user" \
     -H 'accept: application/json' \
   ```

```
  -H 'Content-Type: application/json' \
  -d '{
    "username": "'$ACC_OWNER1_USERID'",
    "old_password": "'$ACC_OWNER1_ASSIGNEDPASS'",
    "new_password": "'$ACC_OWNER1_USERPASS'"
  }'
```

Password must be 15–128 characters long, include uppercase, lowercase, digits, and special characters, and must not contain the username

3. Login to the ACC using owner credentials and obtain a token to perform authentication and authorization using `login API`.

```
response=$(curl -k -X 'POST' \
https://$ACC_IP:$ACC_PORT/api/user/token \
-H 'accept: application/json' \
-H 'Content-Type: application/json' \
-d '{
  "username": "'$ACC_OWNER1_USERID'",
  "password": "'$ACC_OWNER1_USERPASS'"
}')
OWNER_TOKEN=$(echo $response | jq '.access_token' | tr -d '"')
```

The `$OWNER_TOKEN` will be used for authentication and authorization in subsequent steps.

4. Check for available resources that are assigned to the owner using [assigned resources API]()

```
curl -k -X GET \
    "https://$ACC_IP:$ACC_PORT/api/resource/pkgs" \
    -H "Authorization: Bearer $OWNER_TOKEN" \
    -H 'Content-Type: application/json'
```

The owner must ensure that the CPC and logical partitions intended for appliance installation are assigned to them by the ACC admin.

5. Upload appliance image

Upload the image to ACC and set its attributes.

For instance, the owner sets the `min_ifl` attribute to 1, which means that at least one IFL is required to activate the image.

```
curl -k -X 'POST' \
  "https://$ACC_IP:$ACC_PORT/api/images" \
  -H "Authorization: Bearer $OWNER_TOKEN" \
  -H "Content-Type: multipart/form-data" \
  -F "data=@${IMG_LOC}" \
  -F "image_type"="image" \
  -F "min_ifls=2" -F "min_memory=8192"
```

For new appliance installations, the `image_type` must be set to `'image'`. This indicates that the image located at `$IMG_LOC` will completely wipe the disk and install a fresh appliance.

.

6. You can retrieve image details from ACC using the Image Info API.

```
curl -k -X 'GET' \
  "https://$ACC_IP:$ACC_PORT/api/images" \
  -H "Authorization: Bearer $OWNER_TOKEN"
```

To display the output in a readable format, use the `jq` utility.

From the response, you can extract the `image_id` to specify which image to install.

Note: When replacing an existing image in ACC, a name or version conflict may occur. To resolve this, delete the existing image using the following API:

```
curl -k -X 'DELETE' \
"https://$ACC_IP:$ACC_PORT/api/images/<image_id>" \
-H "Authorization: Bearer $OWNER_TOKEN"
```

7. You can activate and install the appliance on an assigned LPARs.

- Since LPAR:

```
curl -k --location "https://$ACC_IP:$ACC_PORT/api/cluster/activate" \
  --header "Content-Type: application/json" \
  --header "Authorization: Bearer $OWNER_TOKEN" \
  --data '{
    "$RP1": {
      "image_id": 1,
      "processor_usage": "dedicated",
      "processor_type": "ifl",
      "memory": $MEMORY,
      "cores": $CORES,
      "username": "'$APP_USERID'",
      "password": "'$APP_USERPASS'",
      "hostname": "host1",
      "lpars": [
        {
          "name": "'$LPAR'",
          "execution_action": "default",
          "install": true
        }
      ]
    }
```

- Multiple LPARs as a Cluster:

```
curl -k --location "https://$ACC_IP:$ACC_PORT/api/cluster/activate" \
  --header "Content-Type: application/json" \
  --header "Authorization: Bearer $OWNER_TOKEN" \
  --data '{
    "rp-1": {
      "image_id": 1,
      "processor_usage": "dedicated",
      "processor_type": "ifl",
      "memory": $MEMORY,
      "cores": $CORES,
      "username": "'$APP_USERID'",
      "password": "'$APP_USERPASS'",
      "hostname": "host1",
      "lpars": [
        {
          "name": "'$LPAR1'",
          "execution_action": "default",
          "install": true
        },
        {
          "name": "'$LPAR2'",
          "execution_action": "default",
          "install": true
        }
      ]
    },
    "rp-2": {
      "image_id": 2,
      "processor_usage": "shared",
      "processor_type": "general-purpose",
      "memory": $MEMORY,
      "cores": $CORES,
      "hostname": "host",
      "username": "'$APP_USERID'",
      "password": "'$APP_USERPASS'",
      "lpars": [
        {
          "name": "'$LPAR3'",
          "execution_action": "default",
          "install": true
        }
      ]
    }
  }'
```

In this example, the appliance is installed using `image_id` 1 on the specified LPAR ($LPAR1), on resource package $RP1, with $CORES shared IFLs and $MEMORY MiB of memory.

The `APP_USERID` and `APP_USERPASS` fields represent the appliance management credentials

This command initiates the appliance installation process on ACC, which may take several minutes to complete. For more information on **`execution_action`** and **`install`** parameters, see "Managing Appliance Cluster " on page 80.

You will get a task id. You can use the task id to verify the status of the appliance.

What to do next

Verify the appliance activation status by running the command:

```
curl -k -X GET \
    "https://$ACC_IP:$ACC_PORT/api/tasks/$TASK_ID/status" \
    -H "Authorization: Bearer $OWNER_TOKEN" | jq
```

If appliance activation is successful, then the status will change to "success". If activation fails, a rollback will be performed and lpar goes to "not activated" state.

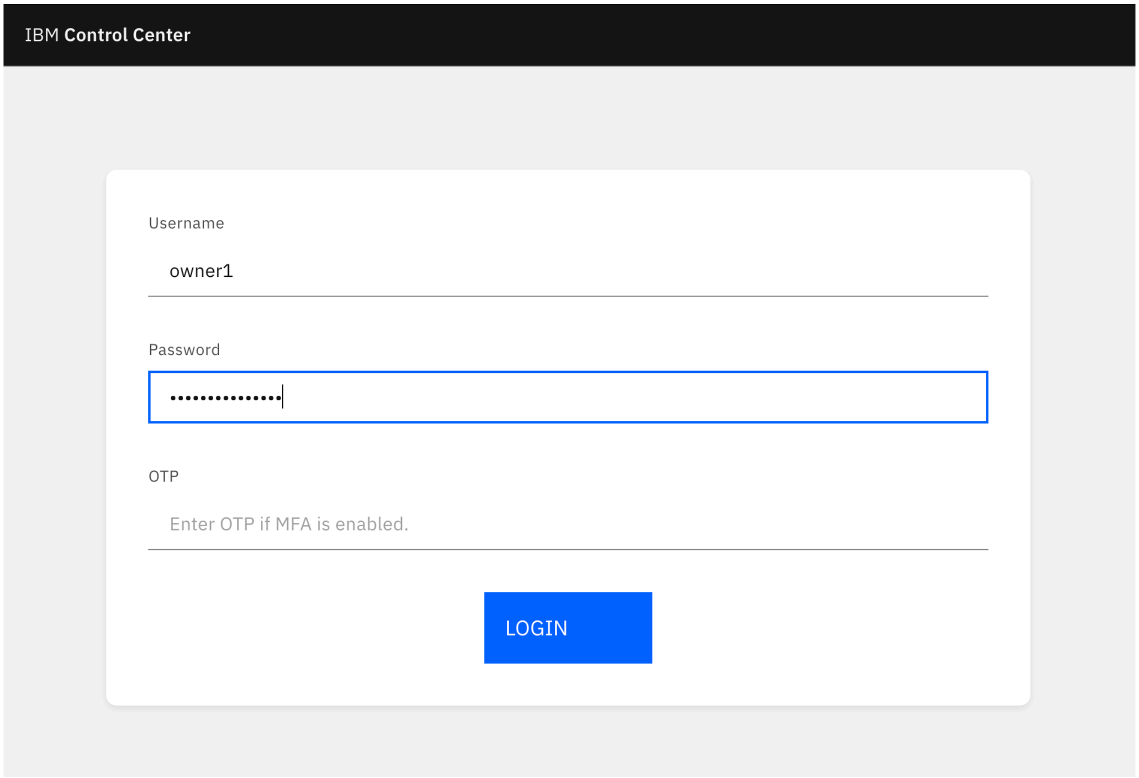
Installing Appliances using ACC UI

The appliance must be installed by the appliance owner. To complete the installation, perform the following steps:

About this task

Procedure

1. Open a browser, and enter *`<ACC_IP_address>`*`:8081`.
2. In the **Login page**, enter your credentials and click **Login**.



IBM Control Center

Username

owner1

Password

•••••••••••••••

OTP

Enter OTP if MFA is enabled.

Figure 4. Login screen

If the ACC administrator has enabled two-factor authentication (2FA), use the time-based one-time password (TOTP) generated by your authentication app. If 2FA is not enabled, this value will be ignored.

The list of LPARs that are assigned by ACC-admin to appliance-owner are displayed in **Managed LPARs** window.

3. Click **Image Catalog** and then click **Upload image**

   If you are want to select an existing image, then you can select one from the catalog.

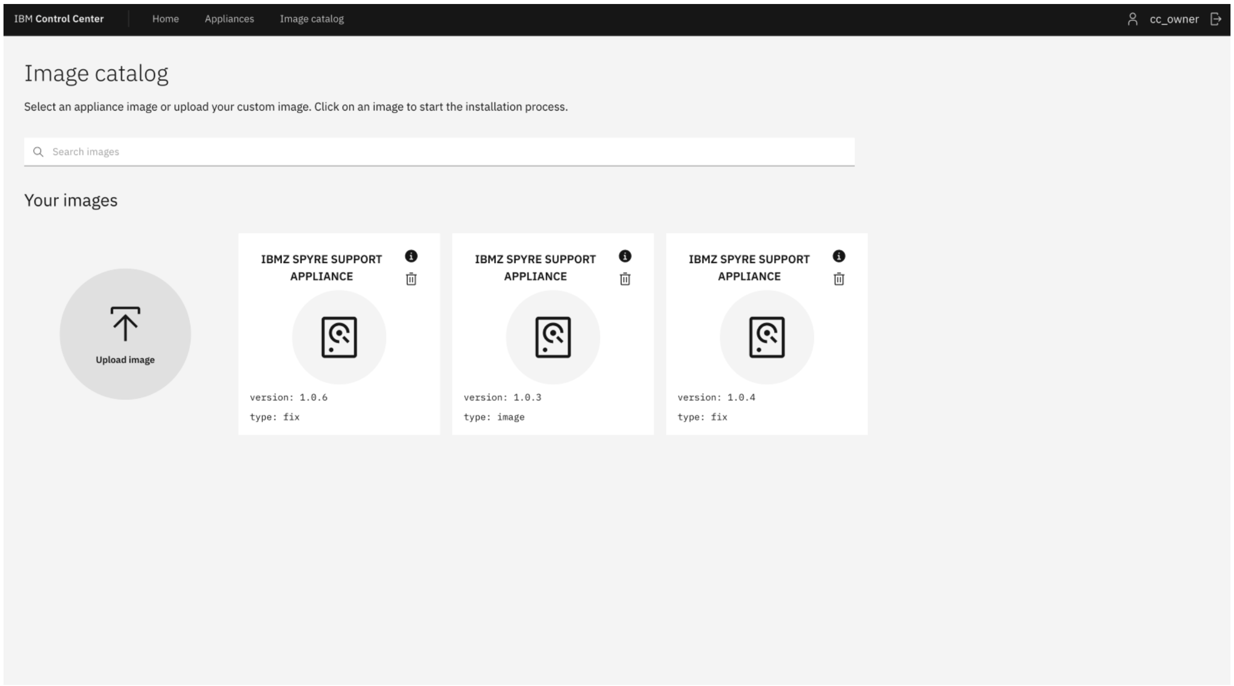


Image catalog

Select an appliance image or upload your custom image. Click on an image to start the installation process.

IBMZ SPYRE SUPPORT APPLIANCE
version: 1.0.6
type: fix

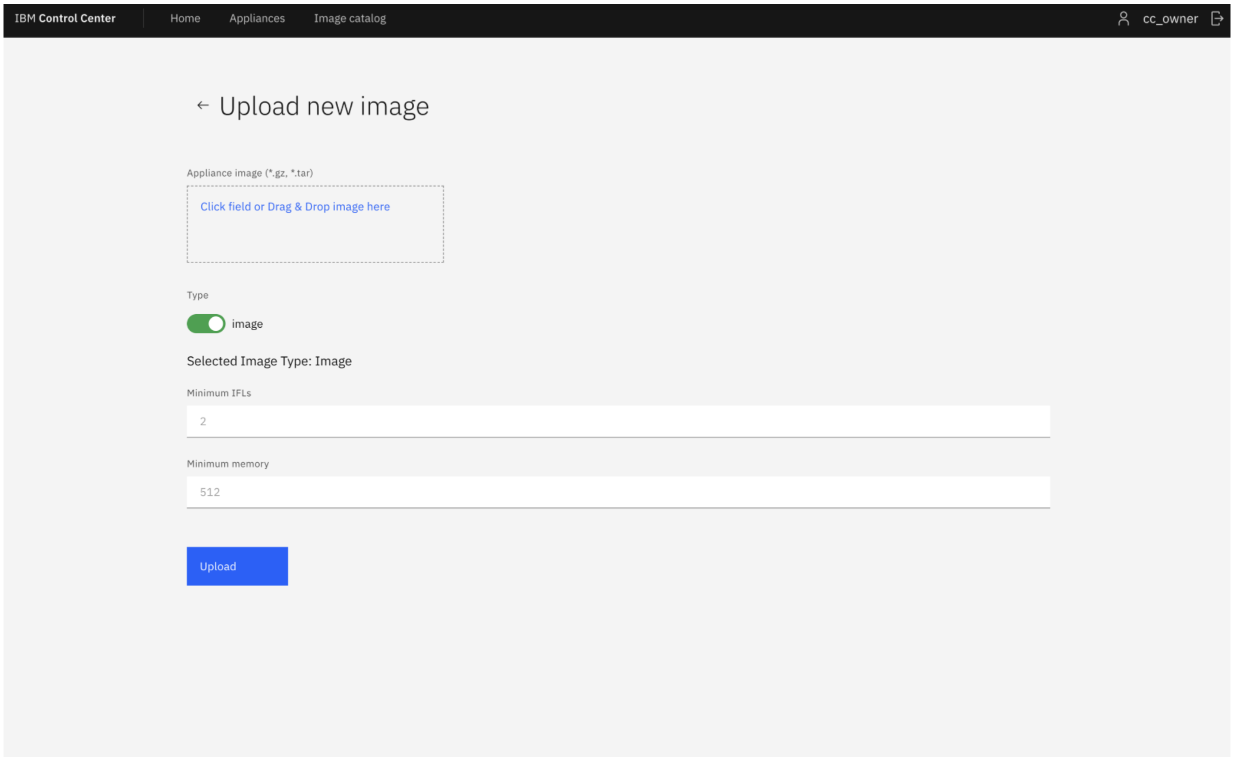
IBMZ SPYRE SUPPORT APPLIANCE
version: 1.0.3
type: image

IBMZ SPYRE SUPPORT APPLIANCE
version: 1.0.4
type: fix

4. Upload the appliance and select the relevant fields

   - Type: Image: Formats the disk and then performs a fresh installation. Fix: The image is considered as an update bundle. ACC performs an update and disk is not formatted.
   - Minimum IFLs : The minimum number of IFLs or CPs that the appliance using the uploaded image file must use.
   - Minimum memory: The minimum memory size in MiB that the appliance using the uploaded image file must use.

   Note: This field must be completed by the appliance owner. It ensures that if the appliance owner attempts to start the appliances with fewer IFLs or GPs than the specified value, ACC will prevent the operation from proceeding.



Appliance image (*.gz, *.tar)

Click field or Drag & Drop image here

Type

image

Selected Image Type: Image

Minimum IFLs

2

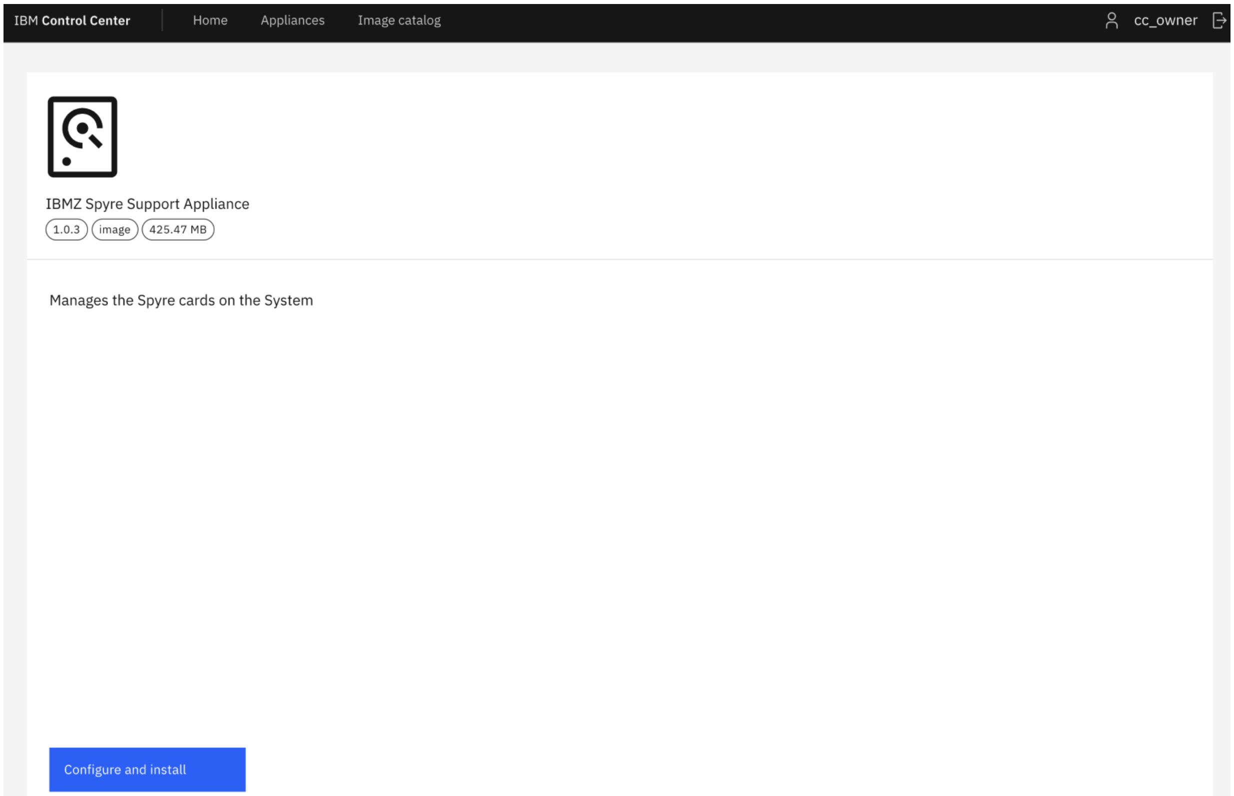
Minimum memory

512

5. Click **Upload**

   Note: If the image file is large and the connection between the control node and ACC has limited bandwidth, the upload step may time out. To verify whether the upload was successful, compare the size of the image on ACC with the original file on the control node. You can do this by clicking on the individual image entry in the ACC UI and checking its size.

6. From the catalog, select the image that you want to install and then click **Configure and Install**



IBMZ Spyre Support Appliance

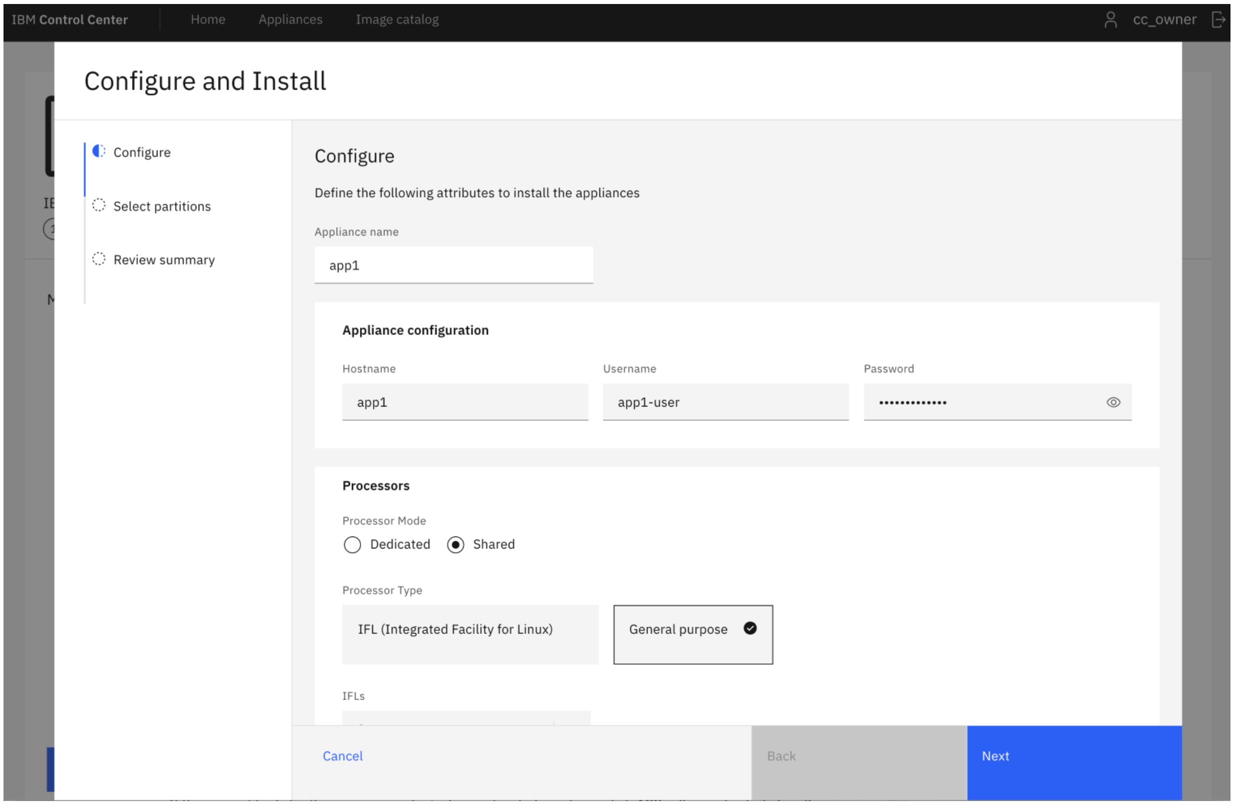
1.0.3 image 425.47 MB

Manages the Spyre cards on the System

The **Configure and Install** screen appears.

7. In the **Configure** tab, enter the following details and then click **Next**.

   - Appliance name: Name of the appliance.
   - Hostname: Hostname of the appliance. This should match the hostname of the SSC Installer as displayed in the HMC user interface.
   - Username: Username of the appliance. This will be set as the username for the SSC Installer as displayed in the HMC user interface.
   - Password: Username of the appliance. This will be set as the password for the SSC Installer as displayed in the HMC user interface.
   - Processor Mode: Select the processor mode as per your requirement.
   - Processor Type: Select the processor type as per your requirement.
   - IFLs: Number of cores (IFLs or CPs) assigned to the appliance's LPAR.
   - Memory: Size of memory in MiB assigned to the appliance's LPAR.



Configure and Install

Configure

Define the following attributes to install the appliances

Appliance name: app1

Appliance configuration

Hostname: app1

Username: app1-user

Password: ••••••••••••••

Processors

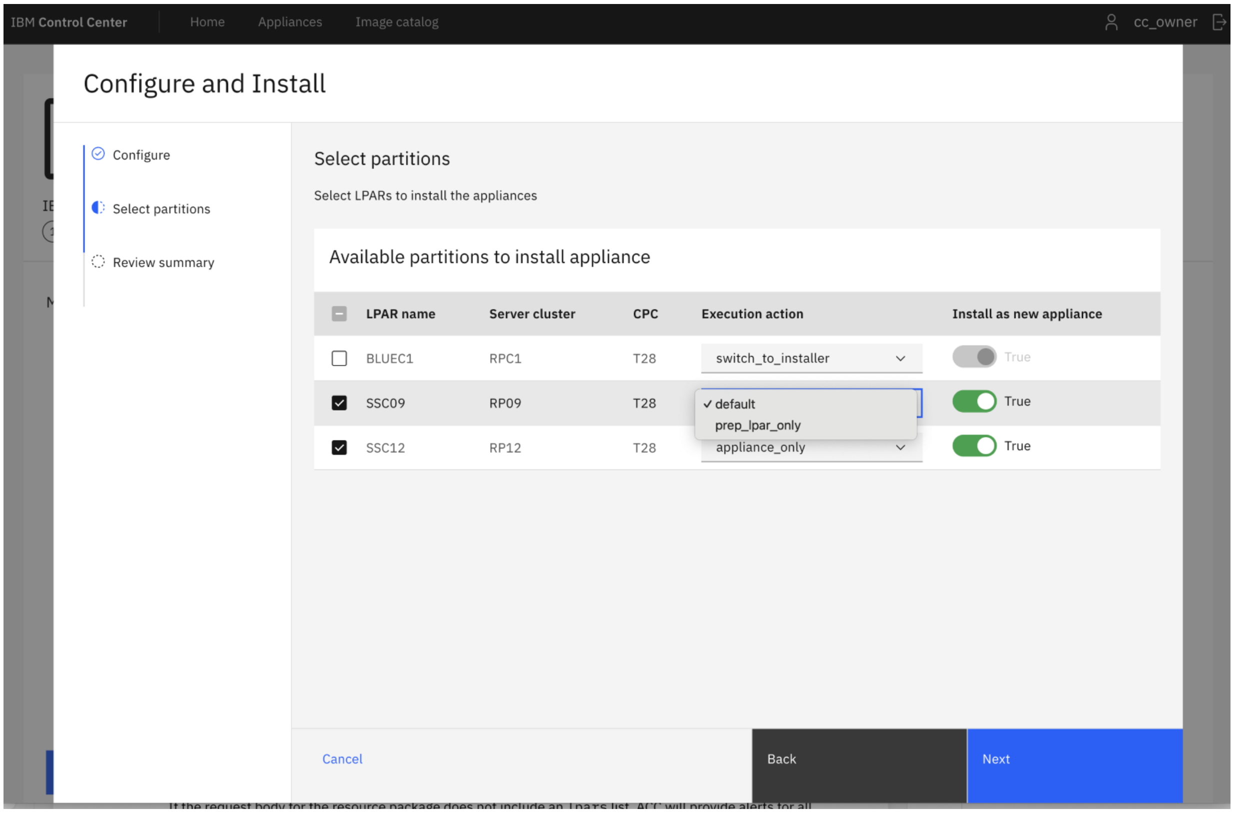
Processor Mode: ○ Dedicated ● Shared

Processor Type: IFL (Integrated Facility for Linux) | General purpose ✓

IFLs

Select partitions tab appears.

8. Select the LPARs on which you want to install image. You can select multiple LPARs and then click **Next**.



Configure and Install

Select partitions

Select LPARs to install the appliances

Available partitions to install appliance

	LPAR name	Server cluster	CPC	Execution action	Install as new appliance
□	BLUEC1	RPC1	T28	switch_to_installer	True
☑	SSC09	RP09	T28	✓ default prep_lpar_only	True
☑	SSC12	RP12	T28	appliance_only	True

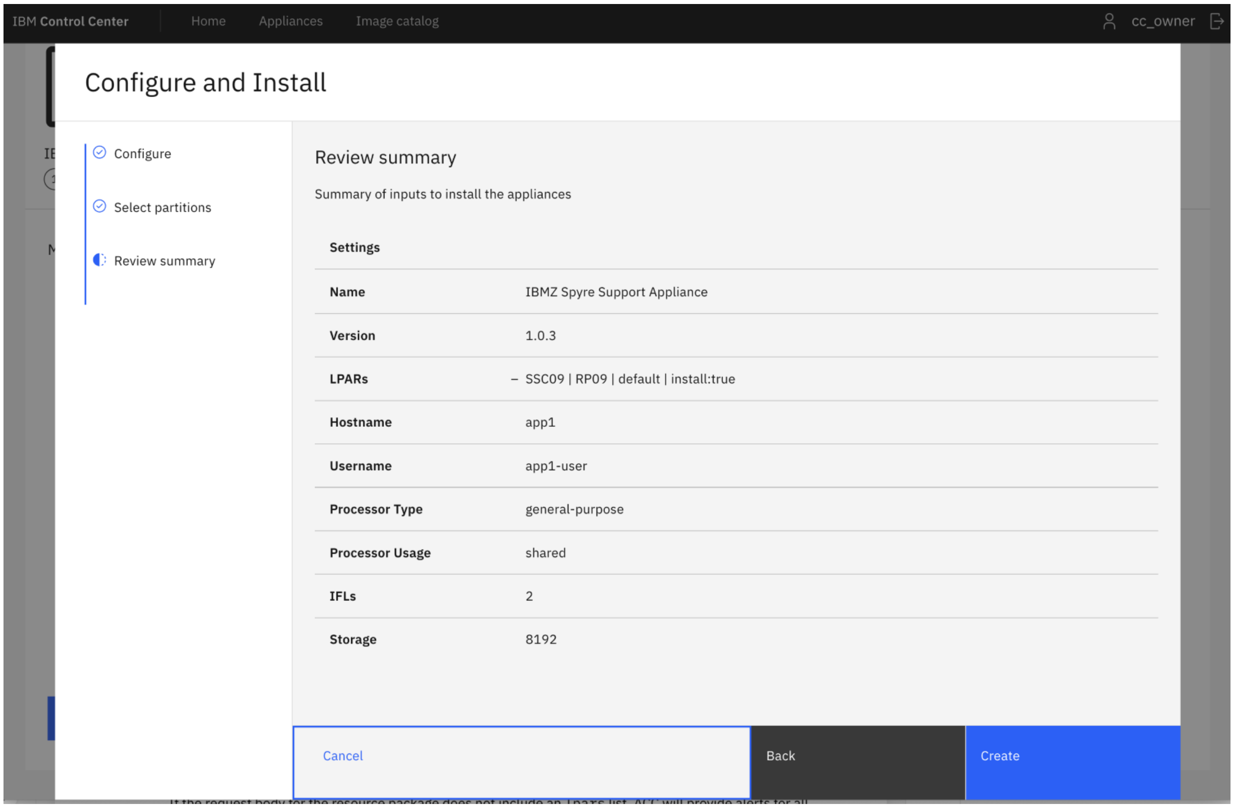
If the request body for the resource package does not include an lpars list, ACC will provide alerts for all

Execution action	Applicable to LPAR	Install as new appliance	Applicable to ACC mode	Resulting ACC action
default	Deactivated	True	Default	The LPAR will first be brought into SSC Installer mode, during which its profile is updated. A new appliance image is then uploaded and installed onto the disk. After installation, the appliance will boot into SSC mode.
default	Deactivated	False	Default	The LPAR profile is first updated, after which the LPAR is brought up in SSC Installer mode. In this mode, the image on the boot disk is compared against the image intended for installation. If the images do not match, an error is raised. If they match, the boot disk is used to start the appliance. **Note:** In this step, images are not installed, only validated and booted.

Execution action	Applicable to LPAR	Install as new appliance	Applicable to ACC mode	Resulting ACC action
switch_to_installer	Activated as appliance		Default and standalone	When an appliance is running on an LPAR in SSC mode, it can be signaled to switch to SSC Installer mode. In this case, the image file is not used.
appliance_only	Activated as Installer	True	Default and standalone	The appliance image file is uploaded to the LPAR while it is in SSC Installer mode. Once the upload is complete, the appliance is installed, and the LPAR boots into SSC mode using the newly installed image.
appliance_only	Activated as Installer	False	Default and standalone	LPAR that is running in SSC Installer mode will be checked to see if the boot-disk image matches the image that is used here. If true, then the LPAR is signaled to switch to SSC mode. That is, the appliance starts running. When an LPAR is running in SSC Installer mode, ACC checks whether the image on the boot disk matches the image intended for use. If the images match, the LPAR is signaled to switch to SSC mode, and the appliance starts running.
prep_lpar_only	Deactivated		Default	The LPAR profile will be updated, after which the LPAR will be activated in SSC Installer mode.

The summary page appears.

9. Review the summary and then click **Create**.



Configure and Install

Review summary

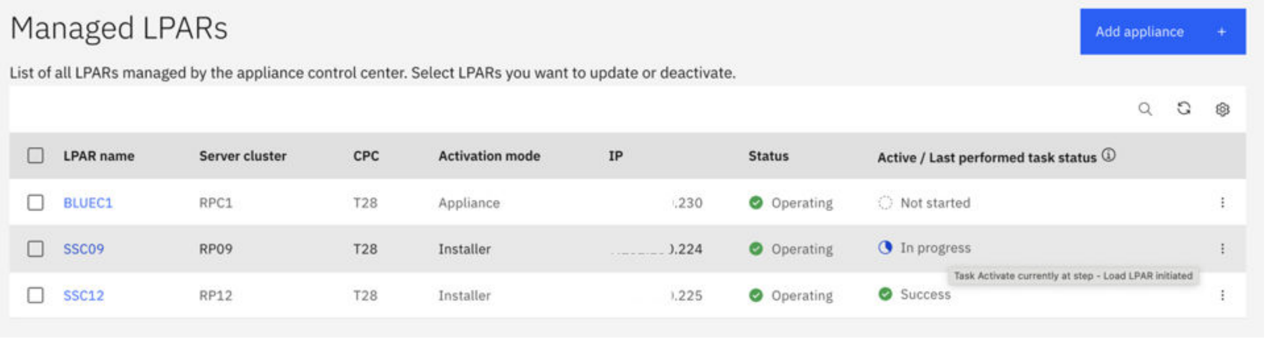
Summary of inputs to install the appliances

Settings	
Name	IBMZ Spyre Support Appliance
Version	1.0.3
LPARs	– SSC09 | RP09 | default | install:true
Hostname	app1
Username	app1-user
Processor Type	general-purpose
Processor Usage	shared
IFLs	2
Storage	8192

If the request is successfully raised in ACC, a pop-up is displayed to confirm it.

What to do next

Go to **Managed LPAR**, and check the status of the LPARs.


Managed LPARs

List of all LPARs managed by the appliance control center. Select LPARs you want to update or deactivate.

□	LPAR name	Server cluster	CPC	Activation mode	IP	Status	Active / Last performed task status ⓘ
□	BLUEC1	RPC1	T28	Appliance	.230	● Operating	○ Not started
□	SSC09	RP09	T28	Installer	).224	● Operating	◔ In progress
□	SSC12	RP12	T28	Installer	).225	● Operating	● Success

Task Activate currently at step - Load LPAR initiated

Chapter 5. Activating and Deactivating an appliance

You can activate and deactivate an appliance by using APIs and UI.

The topics that are covered in this section are:

- “Activating an Appliance using API” on page 45
- “Deactivating Appliance using UI” on page 46

Activating an Appliance using API

Procedure

You can activate a pre-installed LPAR by running the following API:

```
curl -k --location "https://$ACC_IP:$ACC_PORT/api/cluster/activate" \
  --header "Content-Type: application/json" \
  --header "Authorization: Bearer $OWNER_TOKEN" \
  --data '{
    "$RP1": {
      "image_id": 1,
      "processor_usage": "dedicated",
      "processor_type": "ifl",
      "memory": "$MEMORY",
      "cores": "$CORES",
      "username": "'$APP_USERID'",
      "password": "'$APP_USERPASS'",
      "hostname": "host1",
      "lpars": [
        {
          "name": "'$LPAR'",
          "execution_action": "default",
          "install": False
        }
      ]
    }
```

Deactivating an Appliance using API

Procedure

Deactivate multiple appliances.
This command deactivates the specified LPARs in each resource package.

```
curl -k --location "https://$ACC_IP:$ACC_PORT/api/cluster/deactivate" \
  --header "Accept: application/json" \
  --header "Content-Type: application/json" \
  --header "Authorization: Bearer $OWNER_TOKEN" \
  --data '{
    "rp-1": {
      "lpars": [
        "$LPAR1",
        "$LPAR2"
      ]
    },
    "rp-2": {
      "lpars": [
        "$LPAR3"
      ]
    }
  }'
```

Using this command, the appliance owner can deactivate LPARs in resource packages `rp-1` and `rp-2`. The API provides flexibility to target specific LPARs within each resource package for deactivation.

Note: This API is supported only in default mode, as ACC must communicate with the HMC to deactivate the LPARs.

Activating an Appliance using UI

You can activate a pre-installed appliance by following the instructions provided in this topic.

Procedure

1. To activate a pre-installed appliance, following the instructions provided in "6" on page 39
2. Select the image on the catalog that you want to activate and is already installed, and click **Configure and Install**
3. Select the LPAR you want to image to activate.

   The LPAR will have information on the associated boot-disk. ACC will check the information of the image and compare it with the information on the boot-disk
4. In "8" on page 40, set the option "Install as new appliance" to "False".
5. Review the summary and then click **Create**.

Deactivating Appliance using UI

You can activate and deactivate appliance from the **Managed LPAR**s page.

Procedure

1. From the **Managed LPARs** page, select the LPAR, and then click **Deactivate appliance**.
2. A pop-up appears, click **Deactivate**.

Chapter 6. Unlocking the Appliance

The topics in this section provides information on unlocking the appliance using the user interface (UI) and API.

Important considerations

- Only appliance owner tasks are available through the UI. To perform ACC admin tasks, use the APIs.
- The ACC admin must create the appliance owner before proceeding.
- The ACC admin must assign resources to the appliance owner.

The topics that are covered in this section are:

- "Unlocking Appliances using API" on page 47
- "Unlocking Appliance using UI" on page 48

Unlocking Appliances using API

An appliance is locked when ACC does not have the appliance's username and password. As a result, ACC cannot retrieve information from the appliance.

About this task

This situation occurs when an LPAR assigned by the ACC administrator to the appliance owner is already active due to manual activation using the HMC. This can also occur when appliance is switched from appliance to installer mode and vice versa on HMC.

To perform actions such as updating the appliance or gathering logs, the appliance must first be unlocked. Unlocking is done by providing ACC with the appliance's credentials. This can be achieved by using the ACC UI, or using the APIs.

Note: The `/sync/lpars` API currently supports only the synchronization of LPAR status. It does not handle synchronization of LPAR modes (i.e., `appliance` or `installer`).If an LPAR is manually switched between `appliance` mode and `installer` mode using the HMC, this change will not be automatically reflected in ACC.

To synchronize the LPAR mode in ACC:

1. Retrieve the Appliance ID (Quota ID)

   ```
   GET /resource/quotas
   ```

2. Perform Unlock Operation

   ```
   POST /unlock_quota
   ```

Alternatively, you can perform the unlock operation using the ACC UI. This ensures that the LPAR mode is consistent between ACC and HMC.

.

Procedure

1. Get the appliance ID by running the following command:

   ```
   curl -k -X GET \
       "https://$ACC_IP:$ACC_PORT/api/resource/quotas" \
       -H "Authorization: Bearer $OWNER_TOKEN" \
       -H 'Content-Type: application/json' | jq
   ```

2. Run the following command from the control node to unlock an appliance:

```
curl -k -X 'POST' \
  "https://$ACC_IP/api/unlock_quota" \
  -H "Authorization: Bearer $OWNER_TOKEN" \
  -H 'Content-Type: application/json' \
  -d '{
    "ssc_username": "app_username",
    "ssc_password": "app_password",
    "app_id": 1
  }'
```

   where, **app_id** is the ID of the appliance that is retrieved from "1" on page 47

   Note:

   - ssc_username and ssc_password are the credentials configured in the HMC UI during appliance installation on the SSC LPAR of the appliances managed by the ACC. For example, Spyre Support Appliance.
   - These credentials are used by ACC to authenticate and unlock the appliance.

   The appliance with app_id set to 1 is unlocked.

Unlocking Appliance using UI

You can activate the appliance by using the ACC UI.

About this task

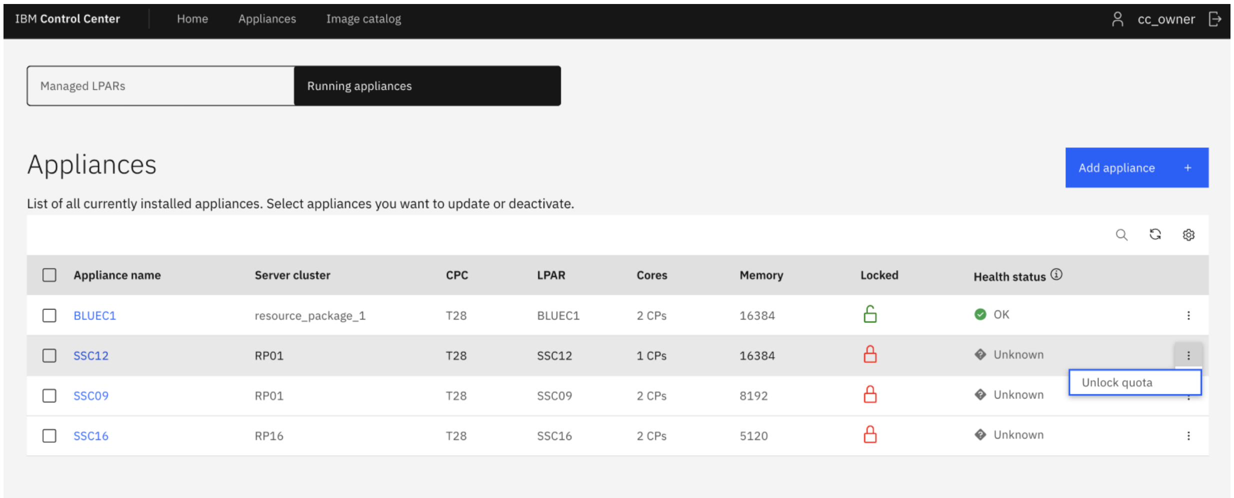
An appliance will be locked if it is already activated using HMC before being managed by ACC. These appliances are locked and are indicated by a red lock symbol.

Appliances activated using ACC are unlocked, indicated by a green unlock symbol.

Note: The ACC's UI can only be used to unlock appliances that are already activated on HMC.

Procedure

1. Go to Running appliances tab and then select the appliance that you want to unlock.
2. To unlock an appliance, click on the three vertical dots located to the right of the appliance and then click **Unlock quota**.

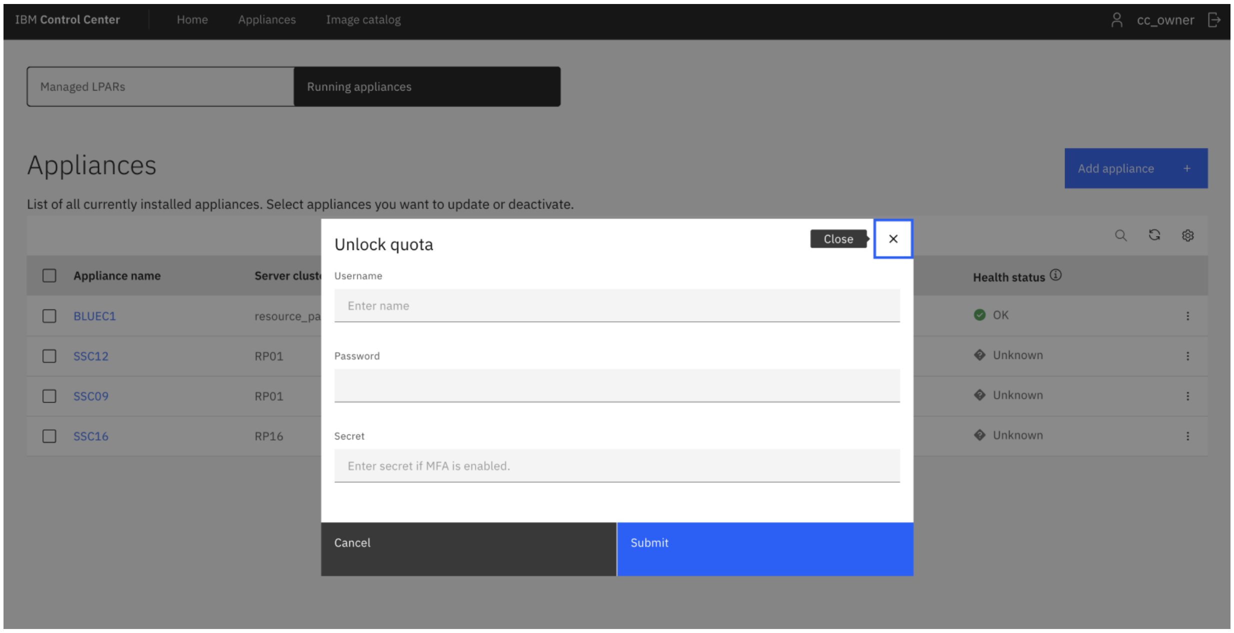


Appliances

List of all currently installed appliances. Select appliances you want to update or deactivate.

☐	Appliance name	Server cluster	CPC	LPAR	Cores	Memory	Locked	Health status	
☐	BLUEC1	resource_package_1	T28	BLUEC1	2 CPs	16384	🔓	OK	⋮
☐	SSC12	RP01	T28	SSC12	1 CPs	16384	🔒	Unknown	⋮
☐	SSC09	RP01	T28	SSC09	2 CPs	8192	🔒	Unknown	⋮
☐	SSC16	RP16	T28	SSC16	2 CPs	5120	🔒	Unknown	⋮

3. Enter the administrator credentials and then click **Submit**.



IBM Control Center | Home | Appliances | Image catalog | cc_owner

Managed LPARs | Running appliances

Appliances

Add appliance +

List of all currently installed appliances. Select appliances you want to update or deactivate.

	Appliance name	Server clust	Health status
□	BLUEC1	resource_pa	OK
□	SSC12	RP01	Unknown
□	SSC09	RP01	Unknown
□	SSC16	RP16	Unknown

Unlock quota

Close ×

Username

Enter name

Password

Secret

Enter secret if MFA is enabled.

Cancel | Submit

Chapter 7. Health Monitoring

Each appliance provides health monitors for various subsystems, and ACC can retrieve the status of these monitors. The health of appliances can be monitored either using the ACC's UI, or by using the APIs.

The topics that are covered in this section are:

- “Health Monitoring Using API” on page 51
- “Health Monitoring Using UI” on page 52

Health Monitoring Using API

ACC can retrieve the status of these monitors by using the APIs.

Procedure

Use the `health` API to check the health of running appliances:

```
curl -k --location "https://$ACC_IP:$ACC_PORT/api/cluster/health" \
  --header "Accept: application/json" \
  --header "Content-Type: application/json" \
  --header "Authorization: Bearer $OWNER_TOKEN" \
  --data '{
    "rp-1": {
      "lpars": [
        "'$LPAR1'",
        "'$LPAR2'"
      ]
    },
    "rp-2": {
      "lpars": [
        "'$LPAR3'"
      ]
    }
  }'
```

This command retrieves health information for the specified appliance cluster. The response includes:

- **`status`**: Indicates the state of individual monitors.
- **`appliance_status`**: Reflects the overall health of each appliance.
- **`overall_status`**: Summarizes the health of the entire resource package.

Status:

Value	Meaning
`OK`	The monitor, appliance, or resource package is functioning normally.
`Degraded`	The monitor, appliance, or resource package requires attention.
`Failed`	The monitor, appliance, or resource package is not functioning as expected.

When ACC sends a health-gathering command to appliances, it waits for each appliance to respond with its health data. If the number of appliances is large, this process can cause ACC to become unresponsive, potentially resulting in timeouts.

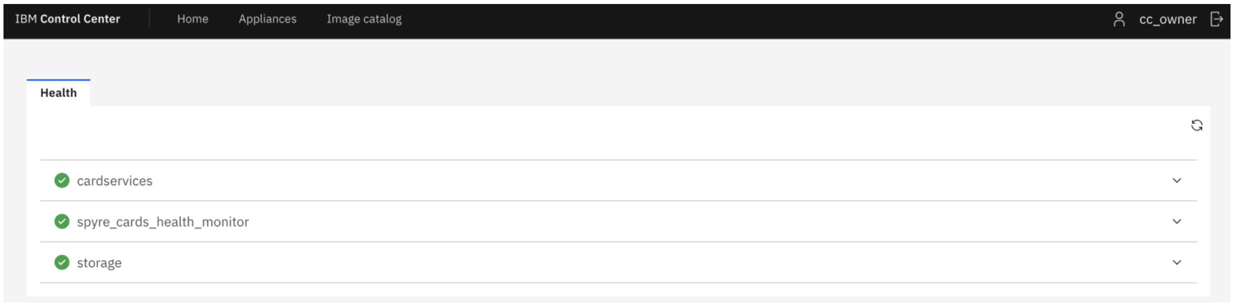
To avoid such issues, it is recommended to target only a small subset of appliances when using the API for health data collection.

Health Monitoring Using UI

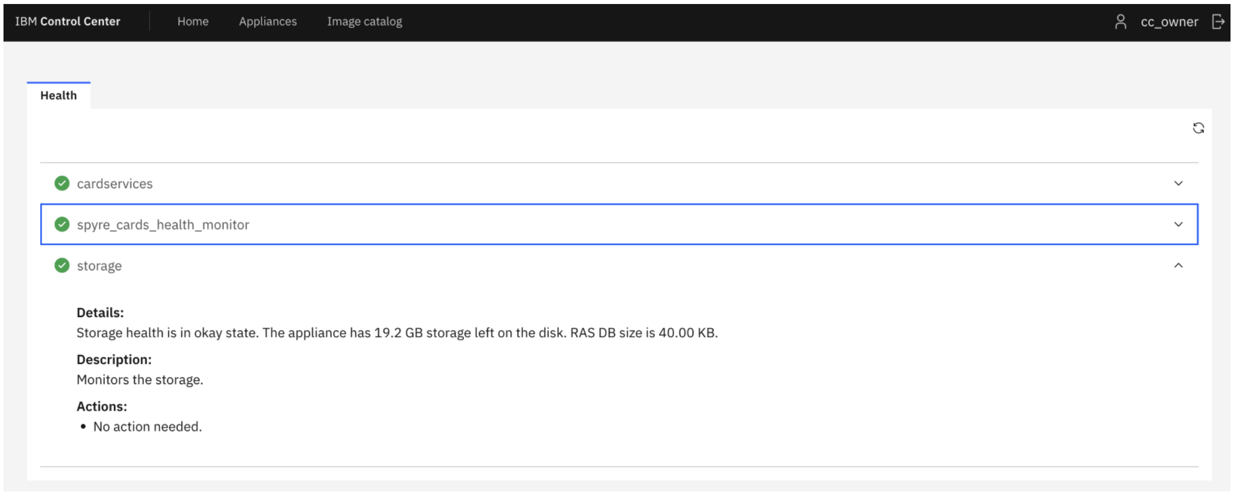
You can monitor the status of appliance by using ACC.

Procedure

1. Login to ACC and then go to **Running appliance** tab.
2. Click on health status of the appliance for which you wish to gather the data.

   Health monitor page appears.
3. To get more details, click on the health monitor.

   Details:
   Storage health is in okay state. The appliance has 19.2 GB storage left on the disk. RAS DB size is 40.00 KB.

   Description:
   Monitors the storage.

   Actions:
   - No action needed.

   Note: If monitors are not available, then the status is displayed as "Unknown".

Chapter 8. Getting Logs of Appliances using Appliance Control Center

Appliances generate log files, that can be used by the Appliance Control Center (ACC) admins, appliance-owners and IBM for debugging problems with the environment or appliances. One can use ACC to create, list and gather the logs.

ACC is a Secure Service Container (SSC)-based appliance, you can use SSC APIs to collect logs.

Note: Some appliances may generate encrypted logs. If the logs are encrypted then they can only be decrypted by IBM.

All log creation commands require an authenticated user.

You can gather logs by using one of the following method:

- “Gathering logs using Ansible Playbook” on page 29
- “Generating an Authentication Token” on page 53
- “Creating Appliance Logs Using ACC Alerts API” on page 53
- “Retrieving a List of Alerts” on page 55
- “Downloading Logs from Appliances” on page 55

Generating an Authentication Token

Generate a token to authenticate as an appliance owner to use ACC APIs.

Procedure

1. Use the token creation API with `curl`.

```
response=$(curl -k -X 'POST' \
  "https://$ACC_IP:$ACC_PORT/api/user/token" \
  -H 'accept: application/json' \
  -H 'Content-Type: application/json' \
  -d '{
    "username": "'$ACC_OWNER1_USERID'",
    "password": "'$ACC_OWNER1_USERPASS'"
  }')
```

2. Extract the token from the JSON response by using `jq`.

```
OWNER_TOKEN=$(echo $response | jq ' .access_token' | tr -d "\"")
```

This `$OWNER_TOKEN` is for authentication and authorization. It is valid for a limited time and must be regenerated after expiration.

Creating Appliance Logs Using ACC Alerts API

Appliances can generate diagnostic logs when triggered by alerts created through ACC APIs.

Available API Endpoints

You can create alerts at the following levels:

API	Description
POST /api/alerts	All appliances in all resource packages
POST /api/alerts/resource/pkgs/<resource_pkg>	All appliances in a specific resource package
POST /api/alerts/resource/pkgs/<resource_pkg>/lpars/<lpar_name>	A specific appliance (LPAR) in a resource package

To create an alert, the appliance owner sends a POST request to the appropriate endpoint.

Example: Create Logs for All Appliances

```
curl -k -X POST \
  "https://$ACC_IP:$ACC_PORT/api/alerts" \
  -H 'Content-Type: application/json' \
  -H "Authorization: Bearer $OWNER_TOKEN" \
  -d '{
    "reason": "Alert for testing",
    "diag_info": "concurrent"
}'
```

reason

The `reason` field is specified by the user and serves as an identifier for the logs. It can contain textual information about the issue that prompted log creation.

diag_info

The optional `diag_info` field defines how diagnostics are collected. When set to `concurrent` (which is the default operation, and the only one supported by ACC), logs are gathered without impacting appliance operations.

download_logs

The `download_logs` field is optional. By default, it is false.

If set to `false`:

- ACC sends an alert to the appliance.
- ACC waits for the appliance to prepare the logs for the alert.
- ACC returns a `success` status to the appliance owner with the metadata of the alert. For example, uuid.
- The appliance owner can later download the logs by using a separate API. For more information, see "Downloading Logs from Appliances" on page 55

If set to `true`:

- ACC sends an alert to the appliance.
- ACC waits for the appliance to finish generating the logs.
- ACC collects the logs from the appliance.
- ACC forwards the logs to the appliance owner.

⚠ **Attention:** Use `"download_logs": true` cautiously, especially for large log files, as it may lead to timeouts.

Note: The `download_logs: true` option is only supported for a single appliance at the appliance or LPAR level

Example: Create Logs for a Specific Resource Package

Send alerts to all appliances in the RESOURCE_PKG1 resource package.

```
curl -k --fail-with-body -X POST \
  "https://$ACC_IP:$ACC_PORT/api/alerts/resource/pkgs/$RESOURCE_PKG1" \
  -H 'Content-Type: application/json' \
  -H "Authorization: Bearer $OWNER_TOKEN" \
  -d '{
    "reason" : "Requesting log-prep of the resource package because ...",
    "diag_info": "concurrent"
}'
```

Example: Create Logs for a Specific Appliance (LPAR)

Send an alert to the appliance running on LPAR. Since `download_logs` is `true`, ACC waits for the appliance to generate logs, collects them, and returns them to the appliance owner.

```
curl -k --fail-with-body -X POST \
  "https://$ACC_IP:$ACC_PORT/api/alerts/resource/pkgs/$RESOURCE_PKG1/lpars/$LPAR1" \
  -H 'Content-Type: application/json' \
  -H "Authorization: Bearer $OWNER_TOKEN" \
  --output "/tmp/$LPAR1.alert.zip" \
  -d '{
    "reason" : "Request log-prep of LPAR because ...",
    "diag_info": "concurrent",
    "download_logs": true
}'
```

When ACC sends an alert command to appliances, it waits for each appliance to generate logs, prepare metadata, and return the information. If the number of appliances is large, this process can cause ACC to become unresponsive, potentially leading to timeouts.

To mitigate this issue, consider targeting only specific logical partitions (LPARs) when gathering logs. This selective approach helps reduce load and improves responsiveness during log collection operations.

Tip: Limit log collection to relevant LPARs to avoid performance bottlenecks in ACC.

Retrieving a List of Alerts

Appliance owners can retrieve a list of alerts using the GET method on the ACC Alerts API. Similar to creating alerts, you can retrieve alerts at different levels:

API	Description

`curl -k -X GET \`
`  "https://$ACC_IP:$ACC_PORT/api/alerts" \`
`  -H "Authorization: Bearer $OWNER_TOKEN"`
	All appliances in all resource packages

`curl -k -X GET \`
`  "https://$ACC_IP:$ACC_PORT/api/alerts/`
`resource/pkgs/pkg1" \`
`  -H "Authorization: Bearer $OWNER_TOKEN"`
	All appliances in a specific resource package (pkg1)

`curl -k -X GET \`
`  "https://$ACC_IP:$ACC_PORT/api/alerts/`
`resource/pkgs/pkg1/lpars/lp1" \`
`  -H "Authorization: Bearer $OWNER_TOKEN"`
	A specific appliance (`lp1`) in a resource package (pkg1)

Downloading Logs from Appliances

To download logs, appliance owners can use the `download_logs` API. This API retrieves the most recent logs and returns them as a `.zip` file, which is organized by resource package and LPAR.

To download logs from an appliance using ACC, the owner must set the following variables before calling the APIs. For example, these variables can be configured as environment variables on the control node.

Variable	Purpose
ACC_IP	IP of ACC

Variable	Purpose
ACC_PORT	Port of ACC responding to the requests, 8081
ACC_OWNER1_USERID	User ID of the owner, assigned by ACC-admin
ACC_OWNER1_USERPASS	Password of the owner
RESOURCE_PKG1	Name of the first resource package (server group)
RESOURCE_PKG2	Name of the second resource package (server group)
LPAR1	Name of the first LPAR used for activities like pulling logs
LPAR2	Name of the second LPAR used for activities like pulling logs
LPAR3	Name of the third LPAR used for activities like pulling logs

Example directory structure:

```
alert_logs/
  └─ resource_pkg1/
       └─ lpar/
          └─ lpar01-uuid.enc
```

To gather logs, the appliance must first be unlocked by providing its username and password to the ACC. The appliance owner can unlock it by running the `unlock_quota` API on the control node.

Example: Download Logs for Multiple Appliances

To download logs for appliances "LPAR1" and "LPAR2" in resource package "RESOURCE_PKG1", and "LPAR3" in resource package "RESOURCE_PKG2", use the following API:

```
curl -k --fail-with-body -X POST \
  "https://$ACC_IP:$ACC_PORT/api/alerts/download_logs" \
  -H 'Content-Type: application/json' \
  -H "Authorization: Bearer $OWNER_TOKEN" \
  -d '{
    "'$RESOURCE_PKG1'": {
      "lpars": [
        "'$LPAR1'",
        "'$LPAR2'"
      ]
    }
    "'$RESOURCE_PKG2'": {
       "lpars": [
         "'$LPAR3'"
      ]
    }
  }' --output "log.file.zip"
```

This command instructs to gather and return the latest logs for the specified appliances.

If the request body for the resource package does not include an `lpars` list, ACC will provide alerts for all LPARs in the resource package. For example,

```
"https://$ACC_IP:$ACC_PORT/api/alerts/download_logs" \
    -H 'Content-Type: application/json' \
    -H "Authorization: Bearer $OWNER_TOKEN" \
    -d '{
            "'$RESOURCE_PKG1'": {
            },
            "'$RESOURCE_PKG2'": {
            }
        }'
```

Chapter 9. Updating Appliances using Appliance Control Center for IBM Z & IBM LinuxONE

You can update an appliance using Appliance Control Center for IBM Z & IBM LinuxONE (ACC) after it has been installed. ACC supports updates using either Fix Central or a locally uploaded appliance image. This topic describes how to perform both update methods using REST APIs or automation scripts.

Important: Updating an appliance replaces its disk contents. Ensure all important configurations and data are backed up. ACC handles export or import automatically during the update.

Appliance Control Center for IBM Z & IBM LinuxONE supports two types of updates:

1. Full Image Replacement (`image`): This update type replaces the entire appliance image and overwrites all disk contents.
2. Software Package Replacement (`fix`): This update type modifies specific software components on the appliance.

ACC and appliances only support sequential updates. In other words, it is **not** allowed to skip versions and each update must be applied in order.

The topics covered in this section are:

- "Checklist" on page 57
- "Updating Appliance by using an Image" on page 58
- "Updating Appliances by using Ansible Playbook " on page 57
- "Rolling Back an Appliance Update " on page 60

Checklist

Ensure the following prerequisites are met before installing the appliance:

- ACC is installed and configured.
- An appliance owner is created in ACC.
- A resource package is allocated to the appliance owner.
- The appliance is started and unlocked using the unlock API.
- For Fix Central updates: The appliance owner has entitlement to download the update image.
- For uploaded image updates: The update image is downloaded to the control node.

Updating Appliances by using Ansible Playbook

You can automate appliance update by using Ansible scripts.

Procedure

You can update the appliance by using one of the following automation options:

Automation option	Script

Using sample scripts	`cd /path/to/sample_scripts`
	`bash 01-info.sh`

Using an ansible playbook	`cd /path/to/other_usecases_ansible`
	`-playbook 01_upgrade_flow.yaml`

Note: Refer to `README.md` in the directory for detailed instructions.

Upgrading Appliances as a Cluster

You can upgrade or update appliances as a cluster using a specific image by following the instructions provided in this topic. In addition to using provided ansible playbooks for updating an appliance, the user can also use REST APIs.

Procedure

Upgrade or update multiple appliances using a specific image.

The `image_id` parameter denotes the image to be used for the upgrade.

```
curl -k --fail-with-body --location "https://$ACC_IP:$ACC_PORT/api/cluster/upgrade" \
--header 'Accept: application/json' \
--header 'Content-Type: application/json' \
--header "Authorization: Bearer $OWNER_TOKEN" \
--output /tmp/upgrade_data.zip \
--data '{
    "rp-1": {
        "lpars": [
            "LP22",
            "LP23"
        ],
        "image_id": 1
    },
    "rp-2": {
        "lpars": ["LPAR4"],
        "image_id": 2
    }
}'
```

Note: You can either update or upgrade using the same procedure.

If no LPARs are specified in the resource package, then all the LPARs that are a part of the resource package will be updated or upgraded. For example, the following API will update the the entire `rp-2` resource package:

```
curl -k --fail-with-body --location "https://$ACC_IP:$ACC_PORT/api/cluster/upgrade" \
--header 'Accept: application/json' \
--header 'Content-Type: application/json' \
--header "Authorization: Bearer $OWNER_TOKEN" \
--output /tmp/upgrade_data.zip \
--data '{
    "rp-2": {
        "image_id": 2
    }
}'
```

Updating Appliance by using an Image

In addition to targetting mulitple LPARs at the same time, you can target a single LPAR for update or upgrade.

Before you begin

- Verify that you meet all the requirements that are defined in the "Checklist" on page 57.
- Appliance-owner has an entitlement to download from Fix Central.
- The appliance is started and unlocked.
- Required credentials and variables are set.

Procedure

1. Set the following environment variables on the admin's control node:

Variable	Description
`ACC_IP`	IP address of ACC
`ACC_PORT`	ACC port (8081)
`ACC_OWNER1_USERPASS`	Appliance owner password
`ACC_OWNER1_USERID`	Appliance owner user ID
`RESOURCE_PKG`	Name of the resource package
`LPAR`	Logical partition name
`IMG_NAME`	Name of the image
`IMG_LOC`	File path of the image
`IMAGE_TYPE`	Used to indicate whether the file is a patch fix or an upgrade. It can be either "fix" or "image"

2. Create an access token:

```
response=$(curl -k -X 'POST' \
"https://$ACC_IP:$ACC_PORT/api/user/token" \
-H 'accept: application/json' \
-H 'Content-Type: application/json' \
-d '{
  "username": "'$ACC_OWNER1_USERID'",
  "password": "'$ACC_OWNER1_USERPASS'"
}')
```

3. Extract the token:

```
OWNER_TOKEN=$(echo $response | jq '.access_token' | tr -d '"')
```

4. Upload the appliance image

```
curl -X 'POST' \
"https://$ACC_IP:$ACC_PORT/api/images" \
-H "Authorization: Bearer $OWNER_TOKEN" \
-H "Content-Type: multipart/form-data" \
-F "data=@${IMG_LOC}" \
-F "image_type"="IMAGE_TYPE" \
-F "min_ifls=1" \
-F "min_memory=512"
```

Note:

image_type parameter	Description	Disruptive	Complete Disk Overwrite
`image`	Replaces the entire appliance with the uploaded image.	Yes	Yes
`fix`	Updates a specific software component of the existing appliance.	Yes or No	No

5. Retrieve the image ID of the uploaded appliance image:

```
curl -k -X 'GET' \
"https://$ACC_IP:$ACC_PORT/api/images" \
-H "Authorization: Bearer $OWNER_TOKEN"
```

You can use the `jq` utility to format the output and extract the `image_id`:

```
curl -k -X 'GET' \
"https://$ACC_IP:$ACC_PORT/api/images" \
-H "Authorization: Bearer $OWNER_TOKEN" | jq
```

6. Use the image ID retrieved in step 5 and update the appliance.

```
curl --fail-with-body -X POST \
"https://$ACC_IP:$ACC_PORT/api/upgrade" \
-H 'Content-Type: application/json' \
-H "Authorization: Bearer $OWNER_TOKEN" \
--output /tmp/upgrade_data.zip \
-d '{
  "'"$RESOURCE_PKG"'": {
    "image_id": 2,
    "lpar": "'$LPAR'"
  }
}'
```

The returned `upgrade_data.zip` is an encrypted backup that can be used for rollback if needed.

Rolling Back an Appliance Update

Restore the appliance to its previous state by using the backup compressed file returned during the update process.

About this task

When you update an appliance by using either Fix Central or an uploaded image, ACC returns an encrypted compressed file (for example, `/tmp/upgrade_data.zip`). This file contains the appliance's configuration and state before the update.

If the update fails or you need to revert to the previous state, you can manually replay this compressed file on a freshly installed appliance.

Procedure

1. Install a fresh appliance image by using ACC.
2. Refer to Secure Service Container User Guide and replay the encrypted compressed file.
3. To restore a configuration file, follow the steps outlined in "Creating an ACC Restore Checkpoint" on page 31.

Chapter 10. API Reference

The topics covered in this section are:

- “Initializing the Appliance Control Center (ACC)” on page 61
- “User Management” on page 62
- “Multi-Factor Authentication” on page 65
- “Certificate management” on page 67
- “Configuration” on page 68
- “CPCs Management” on page 69
- “Sync in between ACC and HMC” on page 71
- “Resource Package” on page 72

Initializing the Appliance Control Center (ACC)

POST /init

This API is used to perform the initial and mandatory setup of the Appliance Control Center (ACC). It sets the operational mode of ACC, which determines how it interacts with other components.

The available modes are:

- Default mode: ACC communicates with the Hardware Management Console (HMC).
- Standalone mode: ACC operates independently and does not communicate with the HMC.

This configuration is exclusive and cannot be changed dynamically after initialization. Therefore, it is critical to select the appropriate mode before invoking this API.

Note: Only the ACC administrator is authorized to invoke this API.

Parameters

username: Enter the username of the admin

password: Enter the password

mfa_enable: true or false

hmc_managed: Set the value to false to select Standalone mode. Set the value to true to select Default mode.

Request body

```
{
  "mfa_enable": true,
  "hmc_managed": true,
  "credentials": {
    "username": "admin",
    "password": "xxxxxxxxx"
  }
}
```

Responses:

200: Successful initialization response with required data.

```
{
  "message": "ACC initialized in default mode",
  "mfa_enabled": false,
  "mode": "default",
  "status": "success",
  "totp_secret": null
}
```

GET /init

This API is used to retrieve metadata related to the initialization status of the Appliance Control Center (ACC).

The response includes the operational mode of ACC, which indicates how it interacts with the Hardware Management Console (HMC). The possible modes are:

- Default Mode: ACC communicates with the HMC.
- Standalone Mode: ACC does not communicate with the HMC

This information is useful for verifying the current configuration of ACC after initialization.

Note: Both ACC administrator and appliance owner are authorized to invoke this API.

Parameters

None

Responses:

200: Successful operation with required response data.

```
{
  "mfa_enabled": "disabled",
  "mode": "default"
}
```

User Management

POST /user/token

This API is used to generate access tokens required for calling other ACC APIs.

It can only be invoked after the user's password has been successfully updated.

If multi-factor authentication (MFA) has been enabled by the ACC administrator, token generation also requires a time-based one-time password (TOTP) from the user's authenticator application. If MFA is disabled, the TOTP input is ignored by this API.

Note: Only authenticated users with updated credentials can call this API.

Parameters

username: Enter the username of the user

password: Enter the password

otp: If MFA is enabled then enter the OTP.

Request body

MFA enabled

```
{
  "username": "cc_admin",
  "password": "cc_passWord@01",
  "otp": "123456"
}
```

Responses:

200: Successful operation with response data.

```
{
  "access_token": "xxxxxx"
  "role": "admin"
}
```

PUT /user

This API is used to update the password of a user in ACC. It can be invoked by both the ACC administrator and appliance owners.

Password validation rules:

- Length: Must be between 15 and 128 characters.
- Allowed Characters: Letters, digits, and special characters (-_#!@$%&?).
- Complexity Requirements:
  - At least one lowercase letter
  - At least one uppercase letter
  - At least one digit
  - At least one special character
- Restrictions: The username must not be part of the password.

This information is useful for verifying the current configuration of ACC after initialization.

Note: No API token will be issued unless this API is called at least once to update the password.

MFA considerations:

- If multi-factor authentication (MFA) is enabled by the ACC administrator, this API requires a time-based one-time password (TOTP) generated by the user's authenticator application.
- Upon successful password update, a new MFA secret will be generated. The user must use this newly issued secret in their authenticator app for all future TOTP generation.

Parameters

username: Enter the username of the user

old_password: Enter the existing password

new_password: Enter the new password

otp: If MFA is enabled then enter the OTP.

Request body:

MFA enabled:

```
{
  "username": "username",
  "old_password": "123_@ld_dummy_pwd",
  "new_password": "123_new_passW@rd",
  "otp": "123456"
}
```

Responses:

200: Successful operation with response data.

```
{
  "message": "Password updated successfully."
}
```

POST /user

This API is used to create a new appliance owner in ACC.

Note: Only the ACC administrator is authorized to invoke this API.

Username validation rules:

- Must be unique
- Length must be between 6 and 16 characters
- Allowed characters: lowercase ASCII letters (a–z), digits (0–9), hyphen (-), and underscore (_)

- Must start with a lowercase ASCII letter
- Must end with a lowercase ASCII letter or digit

Email validation rules:

- Must be unique
- Must contain the `@` symbol
- The domain portion must include both a name and a top-level domain (TLD), separated by a dot

Password validation rules:

- Length must be between 15 and 128 characters
- Allowed characters: letters, digits, and special characters (`-_#!@$%&?`)
- Must include:
  - At least one lowercase letter
  - At least one uppercase letter
  - At least one digit
  - At least one special character
- The username must not be part of the password

Parameters

username: Enter the username of the user

email: Enter the email address

password: Enter the password

role: Currently, only owner role is supported.

Request body:

MFA enabled:

```
{
  "username": "username",
  "email": "john@email.com",
  "password": "12345",
  "role": "owner"
}
```

Responses:

204: Successful operation with no content in response.

GET /user

This API is used to retrieve a list of all appliance owners registered in ACC.

In addition to basic user details, the response includes the timestamp of the last successful login for each appliance owner. This timestamp is provided in Coordinated Universal Time (UTC).

Note: Only the ACC administrator is authorized to invoke this API.

Parameters

None

Responses:

200: Successful operation with response data

```
[
  {
    "username": "username",
    "email": "john@email.com",
    "last_login_ts": 1696849200
  }
]
```

DELETE /user/{owner_name}

This API is used to delete an appliance_owner.

Note: Only the ACC administrator is authorized to invoke this API.

Parameters

owner_name: The username of the appliance-owner.

Responses:

200: Successful operation with required response data.

Multi-Factor Authentication

These APIs manage and enforce multi-factor authentication (MFA) for user accounts. They allow the administrator to enable, disable, and generate admin and owner's secret for TOTP used in the authentication process.

This group of APIs supports:

- Enabling/disabling MFA: Toggle MFA for a given user based on credentials.
- Status: Check the status of two factor authentication.
- secret: Generate secret for admin and owner (such as TOTP).

Note:

1. Before enabling MFA, ensure that the system time is in sync with the UTC.
2. OTP is valid for one minute.

`PUT /mfa/enable`

This endpoint is used to enable multifactor authentication.

Multi-factor authentication (MFA) can be enabled or disabled by the ACC administrator using the `init` API.

Once MFA is enabled via this API, all user passwords will be flagged for mandatory update. It is the responsibility of the ACC administrator to notify appliance owners that they must update their passwords accordingly.

Upon successful execution of this API, the ACC administrator will receive a temporary MFA secret. This secret must be added to the administrator’s authenticator application to generate a time-based one-time password (TOTP). The administrator can then use their credentials along with the TOTP to update their password. After the password is updated, a new MFA secret will be generated.

Additionally, the ACC administrator must generate temporary MFA secrets for each appliance owner to facilitate their password updates. These secrets should be securely shared with the respective owners. Refer to the `POST /mfa/secret/owner` API for this operation.

Note: Only the ACC administrator is authorized to invoke this API.

Parameters

username: Enter the username of the admin

password: It can be enable or disable

Request body

```
{
  "username": "cc_admin",
  "password": "cc_pw"
}
```

Responses:

200: Successful operation with required response data.

```
{
  "message": "Successfully enabled MFA. Please use the temporary secret to update your
password. This will be valid for 30 mins.",
  "secret": "xxxxxx"
}
```

PUT /mfa/disable

This endpoint is used to enable multifactor authentication.

Note: Only the ACC administrator is authorized to invoke this API.

Parameters
None

Responses:
200: Successful operation with required response data.

GET /mfa/status

This endpoint is used to check the status of multifactor authentication.

This API can be called by both ACC-admin and appliance-owner.

Parameters
None

Responses:
200: Successful operation with required response data.

```
{
  "MFAStatus": "enabled"
}
```

POST /mfa/secret/admin

This API is used to retrieve a temporary multi-factor authentication (MFA) secret for the ACC administrator.

It is intended for scenarios in which the ACC administrator has forgotten or lost access to their existing MFA secret.

The API generates a temporary MFA secret. The ACC administrator must add this secret to their authenticator application to generate a time-based one-time password (TOTP). Using their credentials and the TOTP, the administrator can then update their password. After the password is successfully updated, the ACC system will issue a new MFA secret, which must be used for all future TOTP generation.

Note: Only the ACC administrator is authorized to invoke this API.

Parameters

username: Enter the username of the admin

password: It can be enable or disable

Request body

```
{
  "username": "cc_admin",
  "password": "cc_pw"
}
```

Responses:
200: Successful operation with required response data.

```
{
  "message": "Please use the temporary secret to update your password. This will be valid
for 30 mins.",
  "secret": "xxxxxx"
}
```

POST /mfa/secret/owner

This API is used to generate a temporary multi-factor authentication (MFA) secret for an appliance owner.

It is applicable in the following scenarios:

- When MFA has been enabled for ACC by the ACC administrator.
- When an appliance owner has lost access to their existing MFA secret.

Note: Only the ACC administrator is authorized to invoke this API.

Parameters
password: It can be enable or disable

Request body

```
{
  "username": "cc_owner"
}
```

Responses:
200: Successful operation with required response data.

```
{
  "message": "This is a temporary 2FA secret which is valid for 30 mins. Please provide it
to the user cc_owner to update password",
  "secret": "xxxxxx"
}
```

Certificate management

post /certificate/csr

This API is used to manually generate a key and a Certificate Signing Request (CSR) within ACC.

This API returns a CSR to the ACC-admin. The administrator must then submit this CSR to a trusted Certificate Authority (CA) for signing.

Once the signed certificate is obtained, it can be uploaded to ACC using the

```
POST
        /certificate/upload
```

API.

Note: Only the ACC administrator is authorized to invoke this API.

Parameters
None

Request body

```
{{
  "country": "IN",
  "state": "Karnataka",
  "ip": "0.0.0.0"
}
```

Responses:
200: Successful operation with the generated certificate signing request (CSR)

```

-----BEGIN CERTIFICATE REQUEST-----
```

```
     ...QAwWTELMAkGA1UEB...
-----END CERTIFICATE REQUEST-----
```

POST /certificates.upload

This API is used to upload a signed certificate to ACC in PEM format after the Certificate Signing Request (CSR) has been signed by a Certificate Authority (CA).

The signed certificate must correspond to the CSR previously generated using the key and CSR generation API. Once uploaded, ACC will use this certificate for secure communications.

Note: Only the ACC administrator is authorized to invoke this API.

Parameters
None

Responses:
201: Certificate uploaded successfully.

```
{
  "message": "Certificate uploaded and applied successfully"
}
```

Configuration

POST /config/hmcconfig

This API is used to update the Hardware Management Console (HMC) configuration in ACC when operating in default mode.

Key behaviour:

- The HMC access credentials are stored by ACC in an encrypted memory buffer, which is automatically flushed after 24 hours.
- This API must be called by the ACC administrator every 24 hours to refresh the HMC credentials.
- It is strongly recommended that the ACC administrator backs up the ACC configuration before proceeding. Backups can be performed using the SSC APIs or the provided Ansible playbooks.

Synchronization Behavior:

- Invoking this API triggers a synchronization between ACC and the HMC.
- ACC will begin pulling data from the HMC, which may take approximately 2 to 5 minutes to complete.
- During this sync period, some ACC REST API calls may not function correctly. It is recommended to **wait a few minutes** after invoking this API before making further API calls.

Note:

- ACC supports only a single credential for accessing the HMC.
- Only authenticated users with updated credentials can call this API.

Parameters
host: IP address of the HMC

userid: Enter the username that is used to login to HMC.

password: Enter the password that is used to login to HMC.

verify_cert: Currently, only false is supported.

Request body
Certificate verification disabled.

```
{
  "host": "192.168.X.X",
  "userid": "acc_user_on_hmc@email.com",
  "password": "xxxxxx",
  "verify_cert": false
}
```

Responses:
204: Successful operation with no content in response.

GET /config/hmcconfig

This API is used to fetch the current Hardware Management Console (HMC) configurations that is stored in ACC operating in default mode.

Note: Only the ACC administrator is authorized to invoke this API.

Parameters
None

Responses:
200: Successful operation with required response data.

```
{
  "host": "192.168.X.",
  "userid": "acc_user_on_hmc@email.com",
  "verify_cert": false
}
```

CPCs Management

GET /cpcs/hmc-connection

This API is used to verify whether the connectivity between the ACC and the Hardware Management Console (HMC) is functioning correctly.

Note:

- This API is available only when ACC is operating in the default mode.
- Only the ACC administrator is authorized to invoke this API.

Parameters
None

Responses:
200: Successful operation with required response data.

```

  "hmc_id": "hmc1.com",
  "hmc_user": "username",
  "connection-status": "connected"
}
```

GET /cpcs

This API is used to retrieve the list of CPCs known to ACC. ACC maintains CPC names internally, and their source depends on the operational mode:

- In default mode, CPC names are retrieved by querying the Hardware Management Console (HMC).
- In standalone mode, CPC names are manually provided by the ACC administrator using the `POST / cpcs` API.

This API is useful for obtaining information about the CPCs currently available to ACC.

Note: Only the ACC administrator is authorized to invoke this API.

Parameters
None

Responses:

200: Successful operation with required response data.

```
[
  "P000M96",
  "T28",
  "P000F28"
]
```

POST /cpcs

This API is used to manually insert or update information about CPC in ACC when operating in standalone mode.

Since ACC does not have access to the Hardware Management Console (HMC) in standalone mode, it cannot automatically fetch CPC data. Therefore, the ACC administrator must use this API to provide CPC details manually.

This information is essential for other ACC components, such as resource availability checks.

Key considerations:

- Duplicate CPC names within the same request are not allowed.
- This API can also be used to update existing CPC information, such as the number of Integrated Facility for Linux® (IFLs) or memory configuration.

Note: Only the ACC administrator is authorized to invoke this API.

Parameters

None

Request body

```
{
  "cpcs": [
    {
      "cpc_name": "T28",
      "ifls": 0,
      "gps": 108,
      "available_storage": 1024,
      "dpm_enabled": false
    }
  ]
}
```

Responses:

200: Successful operation with required response data.

```
{
  "message": "CPC entries added / updated successfully",
  "no_of_cpcs_added": 1,
  "status": "success"
}
```

GET /cpcs/{cpc_name}/resource

This API is used to retrieve configuration details and available resources for a specified CPC by its name.

It is particularly useful for understanding the resource availability on a CPC and planning how to distribute those resources among appliance owners.

The CPC name must be obtained from the list of CPCs stored in ACC. Refer to the `GET /cpcs` API for retrieving CPC names.

Mode-specific behaviour:

- In default mode, CPC information is sourced from the Hardware Management Console (HMC).
- In standalone mode, CPC information is manually provided by the ACC administrator

Note: Only the ACC administrator is authorized to invoke this API.

Parameters
cpc_name: The name of the CPC whose configuration is to be retrieved

Responses:
200: Successful operation with required response data.

```
{
  "name": "T28",
  "iflNumber": 40,
  "gcpNumber": 40,
  "aapNumber": 10,
  "iipNumber": 10,
  "installedStorage": 1026048,
  "hsaStorage": 2048,
  "customerStorage": 1024000,
  "customerStorageCentral": 512000,
  "customerStorageAvailable": 512000,
  "is_dpm_enabled": true
}
```

Sync in between ACC and HMC

Sync LPARs information on HMC and ACC.

POST /sync/lpars

Use this API to sync the LPARs status in the default mode ACC to that of the HMC. This can be useful in situations when someone performs an LPAR action on HMC and that action should be reflected on the ACC. The sync is usually done automatically by ACC with the HMC. However, in certain situations a manual sync may be required. This APIs will not work with standalone mode ACC. In standalone mode, the ACC-admin will have to manually adjust the status of the LPARs on ACC.

Note: Only the ACC administrator is authorized to invoke this API.

Parameters
None

Responses:
200: Successful operation with no content in response.

```

  {
  "message": "All lpars are up-to-date"
}
```

POST /sync/cpcs

Use this API to sync the CPCs status in default mode ACC to that of the HMC. The sync is usually done automatically by ACC with the HMC. However, in certain situations a manual sync may be required. This APIs will not work with standalone mode ACC. In standalone mode, the ACC-admin will have to manually adjust the status of the CPCs on ACC.

Note: Only the ACC administrator is authorized to invoke this API.

Parameters
None

Responses:
200: Successful operation with required response data.

```

  {
  "message": "CPC configuration imported successfully"
}
```

Resource Package

POST /resource/pkgs

Use this API to create a new resource package and assign it to an existing appliance owner.

A resource package is a collection of LPARs (logical partitions) on a single CPC A resource package cannot span multiple CPCs.

Note:

- A resource package must not include LPARs from a CPC that is already part of another resource package.
- The name of the resource package must be unique.
- A resource package cannot be modified once created; it must be deleted and recreated to make changes.

Invoking this API triggers a synchronization between the ACC and the Hardware Management Console (HMC). During this process:

- ACC begins pulling data from the HMC.
- Synchronization may take several minutes to complete.
- Some ACC REST API calls may not function correctly until synchronization is finished.
- It is recommended to wait a few minutes after invoking this API before making additional ACC REST calls.

The response differs based on the ACC mode that you are running.

Default Mode (ACC with HMC Integration)

In this mode, the resource package creation request communicates with the HMC and returns the following fields:

- `lpars_non_ssc`: LPARs that are already activated but not in SSC mode. These LPARs cannot be used by ACC.
- `lpars_not_activated`: Assigned LPARs that are not yet activated. ACC can activate these LPARs and install appliances on them.
- `lpars_other_than_operating_not_activated`: LPARs that are inactive and not in an operating state may require intervention by the HMC administrator. The administrator should check the LPAR status on the HMC and take appropriate action. For example, if an LPAR is inactive and in a non-operating state, it should be manually deactivated using the HMC UI. The associated resource package must then be deleted and recreated.
- `lpars_sync_scheduled`: SSC-mode LPARs that are already activated and assigned to the resource package. These will be synchronized with the HMC. Even if appliances are already running on these LPARs, they can still be added to a resource package. The appliance owner must invoke the `/unlock_quota` API to register their credentials with ACC.

Standalone Mode (ACC without HMC Integration)

In this mode:

- The resource package creation request does not verify values with the HMC.
- The API returns a success response and adds the LPARs to ACC’s internal list.
- The ACC administrator must first configure these LPARs in the HMC UI and activate them in SSC-installer mode before creating the resource package.
- Once configured, the appliance owner can use these LPARs in ACC.

Note:

- ACC will automatically update the `interfaces` and `boot-info` fields in the API request with values retrieved from the HMC.

- However, for consistency and fault tolerance, it is recommended to provide accurate values in the API request.
- The `boot-info` field will also be updated with values retrieved from the appliance itself.
- Only the ACC administrator is authorized to invoke this API.

Parameters

owner_name: The Resource package owner's name

name: The name of the resource package.

ifls: Number of IFLs allocated in the resource package.

gps: Number of GPs allocated in the resource package.

memory: Memory size (in MB)

lpars : The list of LPARS to be managed under the resource package.

bootinfo

- disk_id: Disk-id or device-num depending on the disk type.
- is_fcp: Set it to True, if disk type is FCP.
- lun: lun number (reqd. only if is_fcp=True)
- wwpn: wwpn number (reqd. only if is_fcp=True)

interfaces

- vlan_id: Vlan ID of the NIC.
- chpid: Chip ID of the NIC.
- port: port number of the NIC.
- prefix: Prefix of the network.
- ip: IP of the NIC.
- gw: Gateway IP of the network.
- name: Name of the interface.

cpc: cpc name

Request body:

```
  {
  "owner": "owner_name",
  "name": "rp1",
  "ifls": 1,
  "gps": 16,
  "memory": 1024,
  "lpars": [
    {
      "boot-info": {
        "disk_id": "39cc",
        "is_fcp": false,
        "lun": "lun",
        "wwpn": "wwpn"
      },
      "interfaces": [
        {
          "vlan_id": null,
          "chpid": "42",
          "port": 1,
          "prefix": 12,
          "ip": "10.11.16.10",
          "gw": "10.111.161.10",
          "name": "interface-2"
        }
      ],
      "name": "lpar1"
    }
  ],
  "cpc": "A309",
}
```

Responses:

200: Successful operation with no content in response.

```
  {
  "lpars_non_ssc": "[]",
  "lpars_not_activated": "[lpar1]",
  "lpars_other_than_operating_not_activated": "[]",
  "lpars_sync_scheduled": "[]"
}
```

GET /resource/pkgs

This API returns all resource packages known to ACC.

- ACC-admin can retrieve packages assigned to any owner, filter by CPC name, or by package name using query parameters.
- Appliance-owners can only retrieve packages associated with their own account.

The response includes:

- `available_cores`: Number of unused IFLs and GPs in the resource package.
- `available_memory`

To get detailed usage of IFLs, GPs, and memory per active LPAR/appliance, use the

```
GET /resource/quotas
```

API.

Note: Both the ACC administrator and the appliance owner are authorized to invoke this API.

Parameters

owner: Use this API to get resource packages by owner name. Only ACC-admin can specify this query parameter. If specified by an appliance-owner, it is ignored

cpc: Get resource packages with cpc name as a filter.

name: Query resource packages with resource package name as a filter.

Responses:

200: Successful operation with required response data.

```
[
  {
    "owner": "owner_name",
    "name": "resource-pkg-name",
    "ifls": 2,
    "gps": 16,
    "memory": 1024,
    "available_cores": 10,
    "available_memory": 51194880,
    "lpars": [
      {
        "boot-info": {
          "disk_id": "39cc",
          "is_fcp": false,
          "lun": "lun",
          "wwpn": "wwpn"
        },
        "interfaces": [
          {
            "vlan_id": null,
            "chpid": "42",
            "port": 1,
            "prefix": 12,
            "ip": "10.11.16.10",
            "gw": "10.111.161.10",
            "name": "interface-2"
          }
        ],
        "name": "lpar2",
        "status": "operating"
      }
    ],
    "cpc": "A309",
  }
]
```

GET /resource/pkgs/lpars

This API returns all LPARs allocated to either:

- All appliance owners, or
- A specific appliance owner, using query parameters.

This API supports the following optional filters:

- `owner_name`: Filter by the name of the appliance owner.
- `lpar_name`: Filter by the name of the LPAR.
- `resource_package`: Filter by the name of the resource package.

Note: Only the ACC administrator is authorized to invoke this API.

Parameters

owner: Name of the appliance owner.

lpar: Name of the lpar.

resource_pkg_name: Name of the resource package.

Responses:

200: Successful operation with required response data.

```
[
  {
    "cpc": "A309",
    "lpar": "LP01",
    "boot-info": {
      "disk_id": "39cc",
      "is_fcp": false,
      "lun": "lun",
      "wwpn": "wwpn"
    },
    "interfaces": [
      {
        "vlan_id": null,
        "chpid": "42",
        "port": 1,
        "prefix": 12,
        "ip": "10.11.16.10",
        "gw": "10.111.161.10",
        "name": "interface-2"
      }
    ],
    "mode": "appliance",
    "resource_pkg_name": "rp1",
    "status": "Operating",
    "task_id": 1
  }
]
```

DELETE /resource/pkgs/{pkg_name}

Use this API to delete a resource package allocated to a specific appliance owner.

- Appliance Owner: Can delete their own resource package by specifying the resource package name.
- ACC Administrator: Can delete any resource package.

If an ACC administrator deletes an owner, all associated resource packages are automatically removed.

Default Mode

If any LPARs or appliances are active, they must be deactivated before deleting the resource package. Use this option with caution, as all unsaved data will be lost. To override this behavior, the ACC administrator can use the `force` query parameter and set it to `true`. When `force=true` is specified:

- The ACC does not deactivate any LPARs in the resource package.
- These LPARs remain active and accessible through the HMC.

Standalone mode

In standalone mode, the deletion process does not check for active LPARs and deletes the resource package immediately.

Note: Both the ACC administrator and the appliance owner are authorized to invoke this API.

Parameters

pkg_name: Name of the resource package.

owner: Name of the appliance owner. It should be provided when ACC-admin deletes resource package. You can ignore it if the request is sent by an appliance-owner.

force: Can either be true or false. If true, then the resource package is deleted even if there are active appliances within the resource package. This is only used for default mode ACC.

Responses:

200: Successful operation with required response data.

```
  {
  "message": "success"
}
```

GET /resource/quotas

This API is used to retrieve information about resource quotas created for a specific appliance owner.

A resource quota represents an active SSC-mode LPAR or a running appliance.

Default Mode

- When an appliance is activated, its corresponding resource quota is automatically created by ACC.
- A resource quota is also created when an ACC administrator assigns an already active SSC LPAR to a resource package.
- If an SSC LPAR is deactivated, its corresponding quota is automatically deleted by ACC.

Standalone mode

In standalone mode, ACC automatically creates resource quotas for all LPARs in the resource package without checking their state. It is the responsibility of the ACC administrator to ensure that the LPAR is active on the HMC.

You can apply filters such as, LPAR and operating state (installer or appliance).

Note: Only appliance owner is authorized to invoke this API.

Parameters

lpar: Name of the lpar.

mode: operating mode of the lpar.

is_locked: The lock status of the LPAR. The default state is false.

Responses:

200: Successful operation with required response data.

```
  {
  "id": 1,
  "is_locked": true,
  "cores": 1,
  "current_appliance_name": "zSSA",
    "current_appliance_version": "0.9.2",
    "cpc": "A309",
    "lpar": "lpar02",
    "memory": 1024,
    "name": "resource-quota-name",
    "owner": "owner_name",
    "pkg": "rp1",
    "boot-info": {
      "disk_id": "39cc",
      "is_fcp": false,
      "lun": "lun",
      "wwpn": "wwpn"
    },
    "interfaces": [
      {
        "vlan_id": null,
        "chpid": "42",
        "port": 1,
        "prefix": 12,
        "ip": "10.xx.xx.10",
        "gw": "10.111.xx.10",
        "name": "interface-2"
      }
    ],
    "mode": "appliance"
  }
```

GET /resource/quota/lpars

This API is used to retrieve all active LPARs in use by either all appliance owners or a specific appliance owner.

A resource quota represents an active SSC-mode LPAR or a running appliance.

- Appliance Owners: Can only view information about their own active LPARs.
- ACC Administrators: Can view information about all appliance owners.

If an appliance is in a locked state, ACC does not have credentials for that LPAR or appliance. To unlock, the appliance owner must call the `POST /unlock_quota` API and provide the appliance credentials. These credentials allow ACC to authenticate with the appliance.

Note: Both the ACC administrator and the appliance owner are authorized to invoke this API.

Parameters

owner: Name of the appliance owner.

Responses:

200: Successful operation with required response data.

```
[
  {
    "cpc": "CPC31",
    "is_locked": false,
    "lpar": "LP01"
  }
]
```

POST /unlock_quota

This API is used to unlock a resource quota.

A resource quota for an LPAR becomes locked when an activate or upgrade operation is performed on the same LPAR from the HMC or another ACC server. When this occurs, ACC loses the credentials for the LPAR and cannot gather information about it.

Default Mode

- If already activated LPARs are assigned to a resource package, their resource quotas are created, and the appliances are automatically assigned a locked state.
- The appliance owner must unlock the appliance by calling this API.

- This API call unlocks the quota and updates both boot-info and network-info for the appliance.

Standalone mode

In standalone mode, only the boot-info of the appliance is updated after unlocking.

Once this information is available, ACC can perform actions on the LPAR.

Note: Only the appliance owner are authorized to invoke this API.

Parameters

ssc_username: username for the SSC appliance owner.

ssc_password: Password for the SSC appliance.

app_id: ID of the resource quota to be unlocked

secret: Secret for SSC appliance if the appliance requires 2FA. Required only if MFA is enabled on the appliance.

Request body:

```
{
  "ssc_username": "user",
  "ssc_password": "password",
  "app_id": 1,
  "secret": "xxxxx"
}
```

Responses:

200: Successful operation with required response data.

```
{
  "message": "Resource quota unlocked successfully."
}
```

Managing Appliance Images

POST /images

This API is used to upload an image file to ACC. An image file can be either a complete appliance image or a fix bundle. This is determined by the `image_type` property in the request body.

The image file must be in binary format. Ensure that the file originates from a trusted and certified source.

After the image file is uploaded, ACC extracts information from the file and stores it internally. This information can be viewed using the `GET /images` API. Once successfully uploaded, the image can be used for installation or applying updates.

Some updates require a reboot, which ACC performs after applying the update using the `POST /upgrade` API.

Note: Only appliance owner is authorized to invoke this API.

Parameters

min_ifs : Minimum IFLs/GPs required to run the image

min_memory: Minimum memory (in MB) required to run the image

image_type: Type of image, image=full appliance image, or fix=update bundle

data: Image or fix file in binary format

Responses:

200: Successful operation with required response data.

```
{
  "image_id": 1234
}
```

GET /images

This API is used to retrieve a list of all image files uploaded to ACC.

This API returns information about all image files belonging to an appliance owner. The details include:

- Minimum number of cores and memory required on the appliance to execute the image file.
- Image metadata such as size and type.
- Additional properties relevant to installation or updates.

Note:

- Pay close attention to the `reboot_required` property. This indicates whether the appliance vendor has designated ACC to restart the appliance after applying a fix. If a reboot is required, unsaved data may be lost. Exercise caution when applying such fixes.
- Only appliance owner is authorized to invoke this API.

Parameters
None

Responses:
200: Successful operation with required response data.

```
[
  {
    "name": "SSA_appliance-v0.9.0.tgz",
    "owner": "Owner_name",
    "product_url": "www.ssc-appliance.com/v0.9.0",
    "id": 1234,
    "version": "v0.9.0",
    "image_type": "image",
    "size": 5242880,
    "reboot_required": true,
    "metadata": {
      "shortDescription": "Awesome appliance v0.9.0",
      "longDescription": "This awesome appliance is used to compute the ...",
      "minimumCores": 4,
      "minimumMemoryGB": 8
    }
  }
]
```

GET /images/{image_id}

This API is used to get information about an image file by its image id.

Note: Only appliance owner is authorized to invoke this API.

Parameters
image_id: ID of the image.

Responses:
200: Successful operation with required response data.

```
{
  "name": "SSA_appliance-v0.9.0.tgz",
  "owner": "Owner_name",
  "product_url": "www.ssc-appliance.com/v0.9.0",
  "id": 1234,
  "version": "v0.9.0",
  "image_type": "image",
  "size": 5242880,
  "reboot_required": true,
  "metadata": {
    "shortDescription": "Awesome appliance v0.9.0",
    "longDescription": "This awesome appliance is used to compute the ...",
    "minimumCores": 4,
    "minimumMemoryGB": 8
  }
}
```

DELETE /images/{image_id}

This API is used to delete the image file by its image id.

Note: Only appliance owner is authorized to invoke this API.

Parameters
image_id: ID of the image that is to be deleted.

Responses:
200: Successful operation with required response data.

```
{
  "name": "SSA_appliance-v0.9.0.tgz",
  "owner": "Owner_name",
  "product_url": "www.ssc-appliance.com/v0.9.0",
  "id": 1234,
  "version": "v0.9.0",
  "image_type": "image",
  "size": 5242880,
  "reboot_required": true,
  "metadata": {
    "shortDescription": "Awesome appliance v0.9.0",
    "longDescription": "This awesome appliance is used to compute the ...",
    "minimumCores": 4,
    "minimumMemoryGB": 8
  }
}
```

Managing Appliance Cluster

`POST /cluster/activate`

This API is used to activate a cluster of LPARs..

A cluster is a set of LPARs, which may belong to different resource packages. This API enables simultaneous activation of multiple LPARs as a background task in ACC.

Common Parameters for All LPARs in the Cluster:

- Processor Properties:
  - Usage: `dedicated` or `shared`
  - Type: `ifl` or `general-purpose`
  - Number of cores: Same for all LPARs
- Memory: Same size for all LPARs
- Access Credentials: Username and password for all appliances in the cluster.

  To change credentials for individual appliances, use the appliance APIs and then update them in ACC using `POST /unlock_quota`.
- Hostname Prefix: Applied to all appliances.

Specify `execution_action` for each appliance:

- `prep_lpar_only`: ACC updates the LPAR activation profile using HMC but does not deploy the appliance.

  Mode: Default

  Use: Switch LPAR to SSC installer mode.
- `default`(default value): ACC updates the activation profile using the HMC and installs the appliance.

  Mode: Default

  Use: Switch LPAR to SSC installer mode and install the appliance.
- `appliance_only`: ACC performs actions directly on the appliance LPAR without HMC communication.

Use: Switch LPAR to appliance mode.

- `switch_to_installer`: If an appliance is already running, use this to switch the LPAR to installer mode without HMC communication.

  Use: Prepare appliance for a new image installation.

If `execution_action` fails for any LPAR in a resource package, activation for that package stops. Other packages continue. Fix the issue and resend the API for the failed package only.

Install Parameter (Per LPAR)

- `install=true`: ACC installs the image file on the appliance, wiping the disk and installing a new image.

  ⚠ **CAUTION:** Unsaved data will be lost.

- `install=false`: ACC activates the appliance without installing a new image. Assumes the image is already installed.

Note: Only appliance owner is authorized to invoke this API.

Parameters

hostname : Hostname for SSC network configuration

username: Username for the SSC LPAR

password: Password for the SSC LPAR

cores: Number of cores to be used for activating LPAR. It same for all LPARS

memory: Size (in MB) of memory required to activate LPAR

image_id: ID of the appliance image file

processor_usage: Type of processor usage. The value can be shared or dedicated.

processor_type: The type of processor. The value can be if or general-purpose.

install: Install appliance value.

lpars:

- *name*: The name of LPAR to activate
- *execution_action*: The execution action for an LPAR
- *install*: True for installing appliance image, false for activation only in SSC mode

Request body:

```
{
  "rp-1": {
    "image_id": 2,
    "processor_usage": "shared",
    "processor_type": "ifl,",
    "memory": "8192,",
    "cores": "2,",
    "username": "ssc_appliance_username,",
    "password": "ssc_appliance_p@ssw0rd,",
    "hostname": "ssc_appliance,",
    "lpars": [
      {
        "name": "LP03",
        "execution_action": "default",
        "install": true
      },
      {
        "name": "LP05",
        "execution_action": "prep_lpar_only",
        "install": true
      }
    ]
  },
  "rp-2": {
    "image_id": 2,
    "processor_usage": "shared",
    "processor_type": "ifl,",
    "memory": "8192,",
    "cores": "2,",
    "username": "ssc_appliance_username,",
    "password": "ssc_appliance_p@ssw0rd,",
    "hostname": "ssc_appliance,",
    "lpars": [
      {
        "name": "LP0B",
        "execution_action": "default",
        "install": false
      }
    ]
  }
}
```

Responses:

202: Successful operation with no content response.

```
{
  "status": "Request sent",
  "task-id": 1234
}
```

POST /cluster/deactivate

This API is used to deactivate a cluster of LPARs.

A cluster is a set of LPARs, which may belong to different resource packages.

This API enables simultaneous deactivation of multiple LPARs as a background task in ACC.

Note: This API can only be used in default mode.

Parameters

None

Request body:

```
{
  "rp-1": {
    "lpars": [
      "LP38",
      "LPB2"
    ]
  },
  "rp-2": {
    "lpars": [
      "LP01"
    ]
  }
}
```

Responses:

202: Successful operation with no content.

```
{
  "status": "Request sent",
  "task-id": 1234
}
```

POST /cluster/upgrade

This API is used to upgrade a list (cluster) of running/active appliances. This API replaces the disk contents of the currently running appliances with a new appliance image.

Important: Before proceeding, gather and back up the configuration of SSC appliances to prevent data loss.

When you call this API, ACC returns configuration data and other relevant information about the active LPARs. The data is provided as a ZIP file, which contains a directory for each resource package. Inside the directory:

- upgrade-info.JSON: Contains upgrade metadata, such as `upgrade-info`, `image entry` and `task_id`
- `export.data` (encrypted): Contains configuration data for the appliance. This file includes raw export data associated with the upgrade process. It is in binary format and serves as a backup in case the upgrade fails.

Here is an example of '`upgrade-info.json` file:

```
{
  "upgrade-info": {
    "lpar_info": {
      "id": 1,
      "pkg_id": 1,
      "status": "operating",
      "last_sync_time": 1733992705,
      "source_of_change": "Sync",
      "owner": "owner16",
      "name": "SSC16",
      "task_id": 9
    },
    "image": {
      "id": 1,
      "owner": "owner16",
      "name": "zFAB Template Appliance",
      "version": "0.9.0",
      "description": "This image is an awesome image...",
      "min_ifls": 1,
      "min_mem": 512,
      "image_location": "/zFAB-Appliance-Control-Center/images/owner16_zFAB Template
Appliance_0.9.0.gz"
    },
    "boot_info": {
      "id": "0.0.5f46"
    },
    "ip": "9.152.150.232",
    "username": "user",
    "update_trigger_time": 1733992928.7341938,
    "complete_time": 1733992930.6831481,
    "owner_info": {
      "username": "owner16",
      "email": "owner14@ibm.com",
      "role": 1,
      "last_login_ts": 1733992640,
      "state": 1
    },
    "status": "In-progress",
    "api": "Upgrade",
    "export_data_location": "export_data/SSC16_T28_1733992928.7341938.data"
  },
  "task-id": 9
}
```

Keep the ZIP file safe. It can be used by the appliance owner to restore the original appliance version if the upgrade fails. ACC also stores this file internally and uses it during the upgrade process. If ACC cannot continue with the upgrade for any reason, the appliance owner must manually restore SSC appliances using this ZIP file.

When calling this API will tools like `curl`, use the `--output` directive to save the zip file properly.

Note: Only appliance owner is authorized to invoke this API.

Parameters

image_id: Image that is to be installed on to LPAR as a replacement of the existing image.

lpar: List of LPARS to be upgraded.

Request body

```
{
  "rp-1": {
    "image_id": 2,
    "lpars": [
      "LP01"
    ]
  },
  "rp-2": {
    "image_id": 3,
    "lpars": [
      "LP02",
      "LP03"
    ]
  }
}
```

Response
200: Successful operation with required response data.

POST /cluster/health

This API is used to gather the health status of a cluster of appliances. A cluster is a set of appliances, which may belong to different resource packages. This API enables health checks for multiple appliances in parallel.

If any appliance in the request is not part of its resource package, the entire health API call fails.
The API triggers health gathering for the LPARs and returns their status. Internally, ACC activates multiple monitors within each appliance. Each monitor checks the status of different components, such as, Storage, Network and Database. Each monitor responds with one of the following statuses, OK, DEGRADED and FAILED.

`appliance_status`:Shows the health of each appliance. If any monitor reports FAILED or DEGRADED, the appliance status reflects that condition.

`overall_status`: Shows the health status for the entire resource package.

Note:

- Only appliance owner is authorized to invoke this API.
- It is possible for ACC to become unresponsive while waiting for health gathering to complete. To avoid this, it is recommended not to include too many LPARs or appliances in a single health-gathering request.

Parameters
None

Request body

```
{
  "rp-1": {
    "lpars": [
      "A302",
      "A303"
    ]
  },
  "rp-2": {
    "lpars": [
      "B120",
      "B29"
    ]
  }
}
```

Responses:
200: Successful operation with required response data.

```
{
  "rp-1": {
    "A302": {
      "kind": "response",
      "parameters": {
        "storage_monitor": [
          {
            "timestamp": "142817164",
            "status": "OK",
            "details": "Storage is functioning normally",
            "actions": [
              "No action required"
            ]
          }
        ],
        "network_monitor": [
          {
            "timestamp": "3247817164",
            "status": "DEGRADED",
            "details": "Network latency is too high",
            "actions": [
              "Examine network connectivity"
            ]
          }
        ]
      },
      "appliance_status": "DEGRADED"
    },
    "A303": {
      "kind": "response",
      "parameters": {
        "network_monitor": [
          {
            "timestamp": "1747814264",
            "status": "OK",
            "details": "Network if functioning properly",
            "actions": [
              "No actions required"
            ]
          }
        ]
      },
      "appliance_status": "OK"
    },
    "overall_status": "DEGRADED"
  },
  "rp-2": {
    "B120": {
      "kind": "response",
      "parameters": {
        "cpu_monitor": [
          {
            "timestamp": "1723817164",
            "status": "OK",
            "details": "CPUs are functioning normally",
            "actions": [
              "No action required"
            ]
          }
        ],
        "network_monitor": [
          {
            "timestamp": "11127817164",
            "status": "OK",
            "details": "Network latency functioning normally",
            "actions": [
              "No action required"
            ]
          }
        ]
      },
      "appliance_status": "OK"
    },
    "B39": {
      "kind": "response",
      "parameters": {
        "cpu_monitor": [
          {
            "timestamp": "1747317424",
            "status": "OK",
            "details": "CPUs are functioning normally",
            "actions": [
              "No action required"
            ]
          }
        ],
        "network_monitor": [
          {
            "timestamp": "1847817164",
            "status": "FAILED",
            "details": "Network connectivity to DB failed",
            "actions": [
              "Contact IBM for support"
            ]
          }
        ]
      },
      "appliance_status": "FAILED"
    },
    "overall_status": "FAILED"
  }
}
```

Background tasks

GET /tasks/{task_id}/status

This API is used to get status of an ACC task, using the task id

Whenever a long-running task is initiated by a user, ACC executes the task in the background and returns a task ID. This task ID can be used to retrieve information about the ongoing background task in ACC.

Examples of long-running tasks include

- LPAR activation
- LPAR deactivation

Note: Only appliance owner is authorized to invoke this API.

Parameters
task id: Is the task ID.

Responses:
200: Successful operation with required response data.

```
{
  "task-status-code": 2,
  "task-status": "Activation requested",
  "task-id": 1234,
  "reason": "Reason for running the task",
  "task-action": "action",
  "action-status": "status"
}
```

DELETE /tasks/{task_id}

This API is used to delete a task. A task cannot be deleted if it is in progress.

Note: Only the appliance owner is authorized to invoke this API.

Parameters
task id: Is the task ID.

Responses:
200: Successful operation with required response data.

```
{
  "message": "Task deleted successfully"
}
```

Alerts

POST /alerts

This API is used to create error alerts for log gathering on all appliances across all resource packages. You must provide a reason for generating alerts and diagnostic information related to the issue.

Once this API is executed, ACC sends a request to all appliances to start gathering their logs. Appliances may take several minutes to collect logs and store them in the proper format.

After a few minutes, the appliance owner can call `GET /alerts` to check the status of the generated alerts. Once alerts are successfully created, the appliance owner can download the logs using `POST /alerts/download_logs`.

Note:

- Only appliance owner is authorized to invoke this API.
- It is possible for ACC to become unresponsive while waiting for alert generation to complete. To avoid this, do not include too many LPARs or appliances in a single alert-generation request. Instead, use one of the following APIs:
  - POST /alerts/resource/pkgs/<pkg-name>
  - POST /alerts/resource/pkgs/<pkg-name>/lpars/<lpar-name>

Parameters

reason: Provide a reason for generating alert.

diag_info: enter diagnostic information

Request body:

```
{
  "reason": "Appliance is not healthy",
  "diag_info": "concurrent",
  "download_logs": false
}
```

Responses:

200: Successful operation with required response data.

POST /alerts/resource/pkgs/{resource_pkg}

This API is used to create error alerts for log gathering on appliances within a single resource package.

Once this API is executed, ACC sends a request to all appliances to start gathering their logs. Appliances may take several minutes to collect logs and store them in the proper format.

After a few minutes, the appliance owner can call `GET /alerts` to check the status of the generated alerts. Once alerts are successfully created, the appliance owner can download the logs using `POST /alerts/download_logs`.

Note:

- It is possible for ACC to become unresponsive while waiting for health gathering to complete. To avoid this, it is recommended not to include too many LPARs or appliances in a single health-gathering request.
- Only appliance owner is authorized to invoke this API.

Parameters

resource_pkg: Name of the resource package.

reason: Provide a reason for generating alert.

diag_info: enter diagnostic information

Request body

```
{
  "reason": "Appliance is not healthy",
  "diag_info": "concurrent",
  "download_logs": false
}
```

Responses:

200: Successful operation with required response data.

```
{
  "RP01": {
    "LP03": {
      "response": {
        "kind": "instance",
        "properties": {
          "diag-info": "/api/com.ibm.zaci.system/alerts/70a55d4e-a5ec-11f0-
a61c-7aefae8f85ff/diag-info",
          "md5sum": "d01263dea82f31a4c275d8a7c945928e",
          "msgid": "AZIZ0001E",
          "msgtext": "test_alert_for_logs",
          "self": "/api/com.ibm.zaci.system/alerts/70a55d4e-a5ec-11f0-a61c-7aefae8f85ff",
          "size": 24391048,
          "timestamp": 1760109554,
          "type": "error"
        },
        "resource-name": "alerts",
        "resource-version": "1.0",
        "self": "/api/com.ibm.zaci.system/alerts/70a55d4e-a5ec-11f0-a61c-7aefae8f85ff"
      },
      "status": "Success"
    },
    "SSC16": {
      "response": {
        "kind": "instance",
        "properties": {
          "diag-info": "/api/com.ibm.zaci.system/alerts/70a59598-
a5ec-11f0-8179-1e2ef7a558e0/diag-info",
          "md5sum": "84e8fe3d4514550bc1570cf9c5c86814",
          "msgid": "AZIZ0001E",
          "msgtext": "test_alert_for_logs",
          "self": "/api/com.ibm.zaci.system/alerts/70a59598-a5ec-11f0-8179-1e2ef7a558e0",
          "size": 25731096,
          "timestamp": 1760109552,
          "type": "error"
        },
        "resource-name": "alerts",
        "resource-version": "1.0",
        "self": "/api/com.ibm.zaci.system/alerts/70a59598-a5ec-11f0-8179-1e2ef7a558e0"
      }
    }
  }
}
```

GET /alerts/resource/pkgs/{resource_pkg}

This API is used to get the list of error and notification alerts. This API will get the error alerts from all the LPARs, under a specific resource packages.

Note: Only appliance owner is authorized to invoke this API.

Parameters

resource_pkg: Name of the resource package.

Responses:

200: Successful operation with required response data.

```
{
  "RP01": {
    "BLUEC1": {
      "response": {},
      "status": "Success"
    },
    "SSC16": {
      "response": {
        "error": {
          "32a34660-a5eb-11f0-8179-1e2ef7a558e0": "Daily log cycle"
        },
        "notification": {
          "3a10f961-6ad4-4cf2-9325-00fada20b215": "Update bundle successfully created
at: /var/update-bundle/uploads/update_bundle_123.tar",
          "43306c43-9d32-4f98-abd3-55789364383d": "User initiated the appliance update",
          "851c9783-3bfa-43d6-83fc-fb4ed9cbb7fc": "Removing the repository directory: /var/
update-bundle/update_bundle_repo",
          "8acb9531-15bc-4142-b12f-43323cd4e8bf": "User initiated the appliance update",
          "90fa5c5e-be6b-4911-9823-054655bb16cf": "User initiated the execution of pre-
process scripts.",
          "c4d9dbd6-7521-447c-ba13-9699580d6fb6": "User initiated the execution of post-
process scripts.",
          "e11fbe0b-3318-4302-afaa-17261cee00f9": "Executing script: /etc/update.d/
01_post_install_acc_script.sh",
          "febf13d4-d556-4dd2-ab2a-087986b3cea0": "Executing script: /etc/update.d/
01_post_install_acc_script.sh"
        }
      }
    }
  }
}
```

POST /alerts/resource/pkgs/{resource_pkg}/lpars/{lpar_name}

This API is used to create error alerts for log gathering on the appliances within a single resource package.

Once this API is executed, ACC sends a request to all appliances to start gathering their logs. Appliances may take several minutes to collect logs and store them in the proper format.

After a few minutes, the appliance owner can call `GET /alerts` to check the status of the generated alerts. Once alerts are successfully created, the appliance owner can download the logs using `POST /alerts/download_logs`.

You can set the `download_logs` property in the request body:

`download_logs=true`

- ACC sends an alert to the appliance.
- ACC waits for the appliance to finish generating logs.
- ACC collects the logs from the appliance.
- ACC forwards the logs to the appliance owner.

`download_logs=false (recommended)`

- ACC sends an alert to the appliance.
- ACC waits for the appliance to acknowledge the alert.
- ACC returns a success status to the appliance owner, indicating that logs will be generated.
- The appliance owner can later download the logs using `POST /alerts/download_logs`.
- .

Note: A timeout can occur when `download_logs=true`. It is generally safer to set `download_logs=false` and download logs later using a separate API call.

Note: Only appliance owner is authorized to invoke this API.

Parameters

resource_pkg: Name of the resource package.

lpar_name: Name of the LPAR.

Request body

```
{
  "reason": "Appliance is not healthy",
  "diag_info": "concurrent",
  "download_logs": false
}
```

Responses:

200: Successful operation with required response data.

```
{
  "RP01": {
    "LP03": {
      "response": {
        "kind": "instance",
        "properties": {
          "diag-info": "/api/com.ibm.zaci.system/alerts/70a55d4e-a5ec-11f0-
a61c-7aefae8f85ff/diag-info",
          "md5sum": "d01263dea82f31a4c275d8a7c945928e",
          "msgid": "AZIZ0001E",
          "msgtext": "test_alert_for_logs",
          "self": "/api/com.ibm.zaci.system/alerts/70a55d4e-a5ec-11f0-a61c-7aefae8f85ff",
          "size": 24391048,
          "timestamp": 1760109554,
          "type": "error"
        },
        "resource-name": "alerts",
        "resource-version": "1.0",
        "self": "/api/com.ibm.zaci.system/alerts/70a55d4e-a5ec-11f0-a61c-7aefae8f85ff"
      },
      "status": "Success"
    }
  }
}
```

GET /alerts/resource/pkgs/{resource_pkg}/lpars/{lpar_name}

This API is used to get all the error and notification alerts. This API will get the error alerts from a specific LPARs, under a specific resource packages.

Note: Only appliance owner is authorized to invoke this API.

Parameters

resource_pkg: Name of the resource package.

lpar_name: Name of the LPAR.

Responses:

200: Successful operation with required response data.

```
{
  "RP01": {
    "SSC16": {
      "response": {
        "error": {
          "32a34660-a5eb-11f0-8179-1e2ef7a558e0": "Daily log cycle"
        },
        "notification": {
          "3a10f961-6ad4-4cf2-9325-00fada20b215": "Update bundle successfully created
at: /var/update-bundle/uploads/update_bundle_123.tar",
          "43306c43-9d32-4f98-abd3-55789364383d": "User initiated the appliance update",
          "851c9783-3bfa-43d6-83fc-fb4ed9cbb7fc": "Removing the repository directory: /var/
update-bundle/update_bundle_repo",
          "8acb9531-15bc-4142-b12f-43323cd4e8bf": "User initiated the appliance update",
          "90fa5c5e-be6b-4911-9823-054655bb16cf": "User initiated the execution of pre-
process scripts.",
          "c4d9dbd6-7521-447c-ba13-9699580d6fb6": "User initiated the execution of post-
process scripts.",
          "e11fbe0b-3318-4302-afaa-17261cee00f9": "Executing script: /etc/update.d/
01_post_install_acc_script.sh",
          "febf13d4-d556-4dd2-ab2a-087986b3cea0": "Executing script: /etc/update.d/
01_post_install_acc_script.sh"
        }
      }
    }
  }
}
```

POST /alerts/download_logs

This API is used to retrieve the latest timestamped logs and download them as a ZIP file, organized by resource package and LPAR.

When using tools like `curl`, specify the `--output` option to save the ZIP file to a desired location. The files hierarchy looks like the following:

```
alert_logs/
   └─ resource_package/
        └─ lpar/
             └─ LP01-uuid.enc
```

`LP01-uuid.enc`: Contains log data from the LPAR that might be encrypted.

Note: Only appliance owner is authorized to invoke this API.

Parameters

resource_pkg: Name of the resource package.

lpar_name: Name of the LPAR.

Request body

```
{
  "resource-pkg-1": {
    "lpars": [
      "lpar1",
      "lpar2"
    ]
  },
  "resource-pkg-2": {}
}
```

Responses:

200: Successful operation with required response data.

```
alert-info.zip
```

Appliance Restart

PUT /appliance/restart

This API is used to initiate an appliance restart operation.

CAUTION: Any unsaved data will be lost. If the appliance does not restart after invoking the API, the ACC-admin should check the status of the LPAR on the HMC.

Whenever a long-running task is initiated by a user, ACC executes the task in the background and returns a task ID. This task ID can be used to retrieve information about the ongoing background task in ACC.

Examples of long-running tasks include

- LPAR activation
- LPAR deactivation

Note: Only appliance owner is authorized to invoke this API.

Parameters

resource_pkg_name: Name of the resource package.

lpar: Name of the LPAR.

Request body

200: Successful operation with required response data.

```
{
  "resource_pkg_name": "rp1",
  "lpar": "LP01"
}
```

Responses:

200: Successful operation with required response data.

```
{
  "response": "Successfully restarted the system",
  "task": 3
}
```

Chapter 11. Troubleshooting

This topic provides a list of common operational scenarios that are encountered in Appliance Control Center for IBM Z & IBM LinuxONE (ACC) and outlines recommended mitigation steps for each scenario.

Table 2. Troubleshooting scenarios

Scenario	**Mitigation**
Appliances are installed manually	Unlock appliances in ACC
More LPARs are needed by the owner	Delete and re-create resource package
More resources are needed by the owner	Delete and re-create resource package
Admin has lost the password	Reinstall ACC
ACC needs to be moved to another machine	Export configuration, install, and import configuration
ACC cannot reach the appliance	Wait and run the action again
HMC configuration must be changed	In ACC, remove all resource packages and insert new configuration
Non-DPM LPARs in unhandled state	Perform actions on HMC
DPM partitions in unhandled state	Perform actions on HMC
Appliance activation failure	Check the lpar

Getting ACC Information

You can retrieve information about the ACC and its appliance using the following APIs.

Note: You must have the appropriate authentication token for each API.

Retrieve ACC Information

Use the ACC admin token (ADMIN_TOKEN) to get ACC details:

```
curl -K -X 'GET' \
  "https://$ACC_IP:$ACC_PORT/api/init" \
  -H "Authorization: Bearer $ADMIN_TOKEN"
```

Retrieve Appliance Information

Use the SSC appliance token (SSC_TOKEN) to get details about the appliance running ACC:

```
curl -k -X 'GET' \
  "https://$ACC_IP/api/com.ibm.zaci.system/appliance" \
  -H "accept: application/vnd.ibm.zaci.payload+json" \
  -H "zACI-API: com.ibm.zaci.system/1.0" \
  -H "Authorization: Bearer $APP_TOKEN"
```

API Endpoints

To download the OpenAPI specifications, run the following command:

```
curl -k -X 'GET' \
  "https://$ACC_IP:$ACC_PORT/api/docs/cc-api.yaml" \
  -H 'accept: application/json' \
  -H "Authorization: Bearer $TOKEN" --output cc-api.yaml
```

ACC has no access to HMC

To initialize ACC in stand-alone mode, the ACC administrator must use the `init` API call with the `hmc_managed` parameter set to `false`.

In stand-alone mode, ACC cannot communicate with the Hardware Management Console (HMC). As a result, certain actions—such as activating or deactivating appliances—must be performed manually via the HMC user interface.

Once these manual steps are completed, ACC can be used to perform other operations, including:

- Updating appliances
- Gathering logs
- Monitoring appliance health

Appliances are installed manually

If appliances are installed by using HMC, then they must be added to ACC.

Note: ACC automatically syncs appliance information (such as LPAR details) with the HMC. Any manual entries that are made by the ACC-admin in the resource package will be overwritten during this sync.However, changes related to CPCs (for example, if a CPC enters a `not-operating` state) are not automatically synced. Only LPAR-related data is synchronized.

Adding an appliance in default mode

Appliance-owner logs in, checks the resource package and running appliances.

It triggers a synchronization between the HMC and ACC. During the process, the ACC retrieves LPAR information from the HMC.

The appliance owner logs in to ACC using login API.

Appliance owner checks the resource package that is assigned to the owner by using the resource package API.

The appliance owner checks the list of running appliances by using the resource quota API

ACC will automatically sync the appliances' information with the HMC. However, if something changes with the CPCs (for example, the CPC goes into `not-operating` mode), this will not be automatically synced. Only LPAR-related information is synced.

ACC syncs appliance data from HMC, overwriting manual entries.

Adding appliance in stand-alone mode

Login to HMC and check the status of the appliance by observing the "Operating System Messages"

Manual updates to appliance data must be reflected in ACC by deleting and re-creating the resource package

ACC needs to be moved to another machine

- Do a fresh install of ACC on the new machine, and then export/import the configuration. For more information, see "Creating an ACC Restore Checkpoint" on page 31.
- If moving within the same data center, boot the ACC boot disk on the new machine.

ACC cannot reach the appliance

Possible causes:

Network issues between ACC and the appliance.

Appliance is booting or not accepting connections.

Example errors:

```
HTTPSConnectionPool(host='10.10.10.10', port=8081): Max retries exceeded with url:
/api/com.ibm.zaci.system/api-tokens (Caused by ConnectTimeoutError(...))
```

Mitigation:

Verify network connectivity.

Check appliance status by using HMC (Operating System Messages).

HMC configuration must be changed

Before modifying the HMC configuration, all resource packages associated with the previous HMC must be deleted. Following this, no new ACC tasks should be initiated to ensure that no HMC-related operations are in progress.

Once the previous configuration has been cleared, the ACC administrator (ACC-admin) can send the new HMC configurations using the `/hmcconfg` API.

Note: If a new HMC is being introduced, IBM recommends performing a fresh installation of ACC to ensure optimal configuration and stability

Non-DPM LPARs in unhandled state

For ACC to manage an LPAR on a non-DPM (non-Dynamic Partition Manager) machine, the LPAR must be in either the "operating" or "not-activated" state as reported by the HMC. If the LPAR enters a "not-operating" state or any other unsupported state, ACC will not be able to manage it. This means that operations such as syncing, activation, deactivation, and upgrade will not function.

To fix this issue, perform the following:

- Login to HMC and check the status of the LPAR.
- Fix any errors that you might encounter.

Unable to retrieve CPC information

You will not be able to retrieve CPC information if the credentials are incorrect in "4" on page 21 or if a synchronization error occurs. To resolve this issue:

- Enter correct host address and credentials in "4" on page 21.
- Resync the CPC information by using the following API:

  ```
  curl -k -X POST \
      "https://$ACC_IP:$ACC_PORT/api/sync/cpcs" \
      -H "Authorization: Bearer $ADMIN_TOKEN"
  ```

- Resync the LPAR information by using the following API:

  ```
  curl -k -X POST \
      "https://$ACC_IP:$ACC_PORT/api/sync/lpars" \
      -H "Authorization: Bearer $ADMIN_TOKEN"
  ```

  Note: In the case where you need to put the LPARs into the lock state in the resource packages, resync will set `is_locked` to `true`.

- If the appliance mode (installer or appliance) remains unsynced in a running appliance, then unlock the appliance to update the appliance mode. For more information on unlocking the appliance, see "Unlocking Appliances using API" on page 47

Deleting resource package

To delete a resource, run the following command:

```
curl -k -X 'DELETE' \
  "https://$ACC_IP:$ACC_PORT/api/resource/pkgs/${RESOURCE_PKG}?owner=${ACC_OWNER1_USERID}" \
  -H 'accept: */*' \
  -H "Authorization: Bearer $ADMIN_TOKEN" \
  -H 'Content-Type: application/json'
```

In Default mode, ACC prevents deletion of a resource package if there are currently running appliances associated with it. To override this behavior, the user can include the `force` query parameter with a value of `true`. This allows ACC to proceed with the deletion despite the presence of running appliances. For example, you can modify the URI to the following:

```
/api/resource/pkgs/${RESOURCE_PKG}?owner=${ACC_OWNER1_USERID}&force=true
```

Note: Use `&force=true` to explicitly request force deletion.

However, if an appliance within the resource package is actively running a task, ACC will block the deletion request even with the `force` parameter. In such cases, an exception is thrown indicating that a task with the specified task ID is in progress. The deletion can only proceed once the task is completed.

In standalone mode, ACC automatically checks for running appliances on the HMC. The `force` parameter is not required in this mode.

Background Tasks information

Some requests to ACC may initiate long-running background tasks.

To monitor the status of such tasks, users can query the `task` API. For example, to check the status of a specific task, use the following command:

```
curl -k -X 'GET' \
   "https://$ACC_IP:$ACC_PORT/api/tasks/${task_id}/status" \
  -H 'accept: */*' \
  -H "Authorization: Bearer $OWNER_TOKEN"
```

In certain cases, tasks may linger and fail to complete due to unknown reasons. These tasks can be manually deleted using the following command:

```
curl -k -X 'DELETE' \
   "https://$ACC_IP:$ACC_PORT/api/tasks/${task_id}" \
  -H 'accept: */*' \
  -H "Authorization: Bearer $OWNER_TOKEN"
```

Messages

This topic describes the possible output messages displayed in the Appliance Control Center for IBM Z & IBM LinuxONE (ACC).

E00000 - User Exception **User and/or credentials exception.**

Explanation:
Such exception can occur whenever the user enter wrong inputs. In such situation, the user is advised to carefully go through the input values, e.g., in the request body or the URIs.

User response:
Read user guide

Descriptor code:
E00000

E00001 - Password is not Updated **Password of the ACC user is not updated.**

Explanation:
ACC expects every user to update their default password before performing any actions via REST APIs. Therefore, if the password is not updated, the user is advised to update their password using the PUT `/user` API. This is true for both ACC-admin and appliance-owners.

User response:
Update the password and try again

Descriptor code:
E00001

E00002 - Credentials are Invalid **User entered invalid credentials**

Explanation:
ACC user has inserted invalid credentials. The right credentials (username, password and if required the TOTP) are expected.

Descriptor code:
E00002

E00003 - Token is Invalid	**User token is invalid.**

Explanation

The token provided to ACC for a command is not valid. This could be due to multiple reasons. For example:

- The token is corrupted.
- The token is expired.
- An appliance-owner token is used to perform an ACC-admin action and vice versa.

In such situation, the user is advised to re-create the token using the POST /user/token API, or read the ACC user-guide for more information about ACC-admin and appliance-owner roles.

Descriptor code:
E00003

E00004 - ACC Admin not Allowed	**ACC-admin cannot perform this action.**

Explanation

When the ACC-admin sends a command to the ACC that should only be sent by an appliance-owner, the ACC-admin will get this exception.

In such situation, the ACC-admin is advised to read the ACC user-guide for more information about ACC-admin and appliance-owner roles.

Descriptor code:
E00004

E00005 - Appliance Owner not Allowed	**Appliance-owner cannot perform this action.**

Explanation

When an appliance-owner sends a command to the ACC that should only be sent by the ACC-admin, the appliance-owner will get this exception.

In such situation, the appliance-owner is advised to read the ACC user-guide for more information about ACC-admin and appliance-owner roles.

Descriptor code:
E00005

E00006 - Appliance Owner Already Exists	**Appliance-owner name already exists in ACC.**

Explanation:
If an ACC-admin tries to create a new appliance-owner with a name that already exists in the ACC, then this exception is thrown. The ACC-admin must adhere to the requirements of creating new appliance-owners. Refer to the POST `/user` API for more information. Moreover, the list of users can be obtained by the ACC-admin by running the GET `/user` API.

User response:
Change the appliance-owner name.

Descriptor code:
E00006

E00007 - Appliance Owner does not Exist	**Appliance-owner name does not exist in ACC**

Explanation:
If an ACC user tries to use a username that does not exist, then this exception occurs. The user should check for typos in the input, or determine if the user exists in the ACC. The list of users can be retrieved by the ACC-admin using the GET /user API.

User response:
Check the names of the users in ACC, or contact ACC-admin

Descriptor code:
E00007

E00008 - Appliance Owner's Name is not Conformant	**Name of the appliance-owner is not conformant**

Explanation

If the ACC-admin tries to create an appliance-owner, and does not adhere to the requirements, this exception occurs. The appliance-owner name has the following requirements:

- Must be unique.
- Length: 6-16 characters
- Valid characters: lowercase letters, digits, '-', '_'
- Starts with a lowercase letter
- Ends with a lowercase letter or digit

User response

Read the appliance-owner name requirements

Descriptor code:
E00008

E00009 - Appliance Owner's Password is not Conformant	**Internal cleanup failure.**

Explanation

If the ACC-admin tries to create an appliance-owner, and does not adhere to the requirements, this exception occurs. The appliance-owner password has the following requirements:

- Length: 6-128 characters
- Valid characters: letters, digits, special characters (-_#!@$%&?)
- Must not include the username

User response:
Read the password requirements in user guide

Descriptor code:
E00009

E00010 - Appliance Owner's email is not Conformant

Email is not conformant

Explanation

If the ACC-admin tries to create an appliance-owner, and does not adhere to the requirements, this exception occurs. The appliance-owner email has the following requirements:

- Must contain the @ character.
- The domain should have a name and top-level domain separated by a dot.

User response:
Read the email requirements in user guide

Descriptor code:
E00010

E00011 - Owner has Active Appliances

Appliances exists for appliance-owner

Explanation

If an action is performed on the ACC that stipulates no appliances must be active for an owner, this exception occurs. For example, if the ACC-admin tries to delete an appliance-owner, but the appliance-owner has one or more active appliances, then the deletion is not allowed and this exception occurs.

In such cases, the appliance-owner must deactivate all appliances assigned to the appliance-owner before sending the command again. For the specific case of active appliances, either the appliance-owner deactivates them using the ACC, or a user on the HMC manually deactivates them. Afterwards, the ACC-admin should be able to delete the appliance-owner.

User response:
Remove appliances for appliance-owner before performing the action

Descriptor code:
E00011

E00013 - New password same as old

New password is the same as the old one

Explanation:
If the ACC-admin or appliance-owner has to or tries to update their password and the new password is the same as the old password, then this exception is thrown. Refer to the documentation about PUT /user API.

User response:
Change the new password

Descriptor code:
E00013

E00014 - The role assignment is invalid

Internal cleanup failure.

Explanation

User and/or credentials exception

Such exception can occur whenever the user enter wrong inputs. In such situation, the user is advised to carefully go through the input values, e.g., in the request body or the URIs.

User response:
Read user guide

Descriptor code:
E00014

E00015 - Image type is invalid

Invalid image type is assigned

Explanation:
When uploading an image or fix to ACC, the 'image_type' parameter can only be either 'image' or 'fix'. Any other type will throw this error. Refer to the documentation about POST /images API.

User response:
Read user guide

Descriptor code:
E00015

E00106 - ACC is not initialized

ACC mode not initialized

Explanation:

Before using ACC, the ACC must be initialized by the ACC-admin. This initialization contains different aspects like connectivity to HMC, using MFA for login, default or standalone mode of ACC etc. If not initialized, this exception is thrown. To initialize the ACC, refer to the documentation about POST /init API, which must be called by the ACC-admin.

User response:
Initialize ACC mode status using the /api/init endpoint before proceeding.

Descriptor code:
E00106

E00107 - ACC mode has already been initialized **ACC mode has already been initialized**

Explanation:
If the ACC-admin tries to rerun the POST /init REST API command against the ACC, then this exception is thrown. The ACC can only be initialized once, which means that the POST /init API can only be called once. Refer to the documentation of POST /init API

User response:
ACC mode is already set for this ACC instance. Reinitialization is not allowed.

Descriptor code:
E00107

E00108 - HMC connection is not set **HMC connection is not set**

Explanation

Whenever ACC is set up in the default mode, and a command is sent to the ACC that requires connectivity to the HMC, and if the the HMC connection is not configured, this exception is thrown. For example, if the ACC user wants to get information about the status of an appliance and the HMC connectivity is not configured, this exception is thrown. The ACC-admin must therefore, set up the HMC connectivity. For this purpose, refer to the documentation of POST /config/hmcconfig API.

User response:
HMC configurations needs to be set.

Descriptor code:
E00108

E00109 - Operation not allowed **Operation not allowed**

Explanation

If the ACC-admin has initialized the ACC (using the POST /init API) in standalone mode (i.e., ACC cannot send commands to HMC), and an command is sent to the ACC that requires ACC sending commands to the HMC, then this exception is thrown. For this command to work, the ACC must be in default mode of operation. Example of such commands can be:

- Activate an LPAR.
- Get CPC information

In standalone ACC mode, part of the operation must be executed manually on the HMC, and part of the operation must be done on the ACC.

User response:
Operation not allowed in the default mode of ACC.

Descriptor code:
E00109

E00201 - Validation has failed before activation **Validation has failed before activation**

Explanation

When an appliance owner sends a command to ACC running in default mode, in order to activate an LPAR and boot the appliance, ACC performs a series of checks. These checks include:

- Verifying that the LPAR is in the correct status (i.e., not-activated).
- Ensuring that sufficient resources (IFLs, memory) are available to meet the appliance's requirements.

To revisit the LPAR configuration, use the HMC interface or the `GET resource/quota` API. For appliance image details, refer to the `GET/images` API.

User response:
Check the input data and compare with the stored values.

Descriptor code:
E00201

E00402 - Image name already exists in ACC **Image with the same name already exists in ACC**

Explanation

Whenever a new image or fix is added to ACC, ACC compares its name with the names of existing images already uploaded.

If an appliance owner attempts to upload an image or fix that already exists in ACC, an exception is thrown. Duplicate image names are not allowed.

User response:
Either change the image name or delete the image in ACC

Descriptor code:
E00402

E00403 - Image name does not exist in ACC **Image name does not exist in ACC.**

Explanation

Whenever a REST API command is sent to ACC that requires the use of a previously uploaded image, ACC checks whether the specified image exists.

If the image is not found, ACC throws an exception.

To verify available images and their IDs, use the `GET /images` API. This command returns an `id` for each uploaded image, which should be used as the `image_id` in subsequent ACC commands.

User response:
Check list of available images in ACC.

Descriptor code:
E00403

E00405 - Image name is not provided **Image name is required..**

Explanation

The image name must be included in the input provided to the command sent to ACC. If the image name is missing from the input, ACC throws an exception.

User response:
Ensure that image name is provided in the payload.

Descriptor code:
E00405

E00407 - Appliance Update bundle fix match not found. **Failure in extracting fix information.**

Explanation

When an image or fix is uploaded to ACC, the system performs internal validation. One step in this process involves extracting metadata from the image or fix and populating internal structures within ACC.

ACC also checks for specific sections within the uploaded file. If this validation step fails, an exception is thrown.

If an appliance owner receives this exception, they should verify that the image or fix is from IBM and is not corrupted.

If the file was downloaded directly from a trusted source and the error persists, gather relevant ACC logs using the SSC concurrent dump API or the `*_pull_ssc_logs.yaml` Ansible playbook, and contact the IBM Support team.

User response:
Gather relevant ACC logs using the SSC concurrent dump API or the `*_pull_ssc_logs.yaml` Ansible playbook, and contact the IBM Support team.

Descriptor code:
E00407

E00408 - Update bundle metadata match not found in data **Failure in extracting a part of the image.**

Explanation

When an image or fix is uploaded to ACC, the system performs internal validation. One step in this process involves extracting information from the file and populating internal structures within ACC.

ACC also checks for specific sections within the uploaded file. If any required parts or sections are missing, an exception is thrown.

If an appliance owner receives this exception, they should verify that the image or fix is from IBM and is not corrupted.

If the file was downloaded directly from a trusted source and the error persists, collect relevant ACC logs using the SSC concurrent dump API or the `*_pull_ssc_logs.yaml` Ansible playbook, and contact the IBM Support team.

User response:
Gather relevant ACC logs using the SSC concurrent dump API or the `*_pull_ssc_logs.yaml` Ansible playbook, and contact the IBM Support team.

Descriptor code:
E00408

E00409 - Image Upload failed in AC **Image save operation is failing in ACC.**

Explanation

When an image or fix is uploaded to ACC, the system attempts to store the file on its internal disk.

If saving the file fails, ACC throws an exception. This can happen for several reasons, for example, image save fails when the disk is full.

If an appliance owner receives this exception, they should verify that the image or fix is from IBM and is not corrupted.

If the file was downloaded from a trusted source and the error persists, collect relevant ACC logs using the SSC concurrent dump API or the `*_pull_ssc_logs.yaml` Ansible playbook, and contact the IBM Support team.

User response

Check ACC's disk size. If required, delete the images or fixes.

Descriptor code:
E00409

E00500 - Unhandled exception in ACC	**Unhandled exception in ACC.**

Explanation

If an unknown situation occurs in ACC that cannot be interpreted or handled by the system, ACC throws an exception.

This exception may also be triggered by malformed or incorrect inputs provided to ACC using REST APIs.

If the file was downloaded from a trusted source and the error persists, collect relevant ACC logs using the SSC concurrent dump API or the `*_pull_ssc_logs.yaml` Ansible playbook, and contact the IBM Support team.

User response:
Gather relevant ACC logs using the SSC concurrent dump API or the `*_pull_ssc_logs.yaml` Ansible playbook, and contact the IBM Support team.

Descriptor code:
E00500

E00602 - Remote image checksum and size validation failed	**Validation fails for image provided by the remote server.**

Explanation

When an image or fix is pulled by ACC from a remote site (such as IBM Fix Central), ACC performs a series of validations.

These validations include checking the file size and verifying the checksum of the remote image. If any of these validations fail, ACC throws an exception.

If the file was downloaded from a trusted source and the error persists, collect relevant ACC logs using the SSC concurrent dump API or the `*_pull_ssc_logs.yaml` Ansible playbook, and contact the IBM Support team.

User response:
Gather relevant ACC logs using the SSC concurrent dump API or the `*_pull_ssc_logs.yaml` Ansible playbook, and contact the IBM Support team.

Descriptor code:
E00602

E00707 - Appliance not found in ACC	**Appliance is not available.**

Explanation

When a command is sent to ACC to perform an action on an appliance that cannot be found, ACC throws an exception.

For example, this exception is triggered if a user attempts to upgrade an appliance using an ID that does not exist. Similarly, if an appliance quota is missing and an action is initiated against it, the same exception is thrown.

The ACC user should verify the inputs provided in the REST API call. To confirm the existence of appliances or quotas, use the appropriate GET APIs such as `GET /resource/quotas`.

If the file the error persists, collect relevant ACC logs using the SSC concurrent dump API or the `*_pull_ssc_logs.yaml` Ansible playbook, and contact the IBM Support team.

User response:
Gather relevant ACC logs using the SSC concurrent dump API or the `*_pull_ssc_logs.yaml` Ansible playbook, and contact the IBM Support team.

Descriptor code:
E00707

E00708 - SSC Appliance is not reachable in ACC	**ACC cannot reach the SSC appliance.**

Explanation

If ACC needs to access an appliance but cannot reach it, an exception is thrown.

This issue can occur due to several reasons

- The appliance is not active: Check its status using the `GET /resource/quotas` API or using the HMC user interface.
- The appliance is not ready to accept connections: Confirm its readiness by checking its operational status.
- Network connectivity issues: Ensure that the network between ACC and the appliance is functioning correctly. Also, verify firewall and security settings.

If all configurations appear correct and the issue persists, gather logs from the SSC appliance using the `*_pull_ssc_logs.yaml` Ansible playbook, and contact the IBM Support team. Be sure to use the appropriate variables for the SSC appliance.

User response:
Gather logs from the SSC appliance using the `*_pull_ssc_logs.yaml` Ansible playbook, and contact the IBM Support team. Be sure to use the appropriate variables for the SSC appliance.

Descriptor code:
E00708

E00709 - Appliance returned invalid response **SSC appliance returned a malformed response to ACC.**

Explanation

This error occurs when ACC receives an incorrect response from the SSC appliance.

To fix this,

- Verify that the SSC appliance is functioning as expected by running the diagnostic tests provided with the appliance.
- If the appliance passes diagnostics, gather its logs using the `*_pull_ssc_logs.yaml` Ansible playbook. This playbook collects the necessary data for troubleshooting.

User response:
Gather logs from the SSC appliance using the `*_pull_ssc_logs.yaml` Ansible playbook, and contact the IBM Support team.

Descriptor code:
E00709

E00710 - Appliance is blocked with a task **SSC appliance is currently busy with some other task.**

Explanation

This exception occurs when ACC has already initiated a task for a specific appliance and the user sends another command to the same appliance. Tasks such as activation or deactivation cannot be interrupted, and the appliance may reject new commands. These tasks can sometimes take more than 15 minutes to complete.

To fix this,

- Check the list of running tasks using the `GET /tasks` API.
- If the problem persists:
  - Restart the appliance using the `PUT /appliance/restart` API, or
  - Use the HMC to deactivate and then activate the appliance.

  ⚠ **CAUTION:** This approach may result in data loss.
- If everything appears normal, gather the SSC appliance logs using the `*_pull_ssc_logs.yaml` Ansible playbook.

User response:
Gather logs from the SSC appliance using the `*_pull_ssc_logs.yaml` Ansible playbook, and contact the IBM Support team.

Descriptor code:
E00710

E00711 - Appliances already exists for resource package **ACC cannot perform the task because appliances are running.**

Explanation

This exception occurs when ACC cannot perform an action because some appliances are active. For example, if an ACC administrator attempts to delete a resource package while active appliances are running, the command will fail.

To fix this,

- Deactivate the appliances before performing the command. You can use:
  - `POST /cluster/deactivate`

  –
    ```
    POST
        /resource/pkgs/<resource_pkg_name>/
    lpars/<lpar_name>/deactivate
    ```

- If the appliance owner is unavailable to run these commands, the ACC administrator must log in to the HMC and deactivate the appliances manually.
- Once the appliances are deactivated, rerun the command.

User response:
Deactivate the appliances before proceeding

Descriptor code:
E00711

E00802 - Deletion of task failed **Task cannot be deleted.**

Explanation

This exception occurs when a delete request is sent while a task is still in progress.

To fix this,

- Check the status of the task using:

  `GET /tasks/<task_id>/status`
- If the task does not finish after an extended wait:
  - Restart ACC by logging in to the HMC.
  - Deactivate and then activate the ACC LPAR.

    ⚠ **CAUTION:** This approach may result in data loss.
- Retry the delete request after the restart.

User response:
Wait for the task to finish.

Descriptor code:
E00802

E00807 - Activation Profile could not be created **ACC could not create activation profile.**

Explanation

ACC throws this exception when it cannot find the LPAR name in its internal structure or match it with the information retrieved from the HMC.

This issue typically occurs when the LPAR or partition name has been manually changed on the HMC.

User response:
Gather logs from the SSC appliance using the `*_pull_ssc_logs.yaml` Ansible playbook, and contact the IBM Support team.

Descriptor code:
E00807

E00852 - Updating the activation profile of LPAR failed **Updating the profile has failed.**

Explanation

This exception occurs when ACC forwards an LPAR activation request to the HMC, and the HMC responds with an error code. The request includes parameters that modify the LPAR's activation profile. This issue might occur due to following reasons:

- Incorrect values were provided in the REST API request.
- The CPC does not have sufficient resources to fulfill the requested activation profile update.
- The LPAR no longer exists due to manual actions performed on the HMC.

To fix this,

- Review the CPC information on the HMC.
- Adjust resource requests if necessary.

  For example, if the number of requested IFLs exceeds the available IFLs on the CPC, reduce the requested IFLs or contact the HMC/CPC administrator.
- Verify that the LPAR exists and is properly configured.

User response:
Check for the number of If Ls, memory and verify them on HMC.

Descriptor code:
E00852

E00854 - Loading of LPAR task failed **Loading the image on the LPAR has failed.**

Explanation

If the LPAR fails to load during activation, ACC throws an exception. This issue can occur for several reasons:

- The LPAR is not in the correct state. Check the HMC UI for details.
- Disk parameters are incorrect. Review the resource package and verify how disk parameters are defined. Contact the ACC administrator if you suspect an error.
- Network parameters are incorrect. Review the resource package and verify how network

parameters are defined. Contact the ACC administrator if needed.

User response:
If all configurations appear correct, gather the ACC appliance logs using the `*_pull_ssc_logs.yaml` Ansible playbook and contact IBM Support.

Descriptor code:
E00854

E00902 - Precheck of deactivate task failure **Preparing the deactivation tasks failed.**

Explanation

When a user requests ACC to deactivate an LPAR, ACC performs a series of preliminary checks. If any of these checks fail, ACC throws an exception.

This exception typically occurs when the LPAR is in an incorrect or inconsistent state. The LPAR must be in the operating state to proceed.

To verify the LPAR status, use the `GET /resource/quotas` API or check the HMC UI.

User response:
Check input data for deactivation.

Descriptor code:
E00902

E00903 - Deactivate task failed in ACC **Deactivation task has failed.**

Explanation

When a user requests ACC to deactivate an LPAR, ACC forwards the request to the HMC. If the task encounters an error, ACC throws an exception.

To troubleshoot, review the logs and error messages on the HMC.

User response:
If all configurations appear correct, gather the ACC appliance logs using the `*_pull_ssc_logs.yaml` Ansible playbook and contact IBM Support.

Descriptor code:
E00903

E01102 - Switching appliance to installer mode failed **Switching an appliance to installer mode failed.**

Explanation

When a user sends an installation, activation, or upgrade request to ACC for a specific appliance, ACC initiates a series of tasks. One of these tasks involves switching the LPAR to installer mode, which prepares it for the requested operation. If this action fails, ACC throws an exception.

User response:
If all configurations appear correct, gather the ACC appliance logs using the `*_pull_ssc_logs.yaml` Ansible playbook and contact IBM Support.

Descriptor code:
E01102

E01103 - Upgrading to new appliance failed **Installing an appliance has failed.**

Explanation

If the installation request for a new appliance fails, ACC throws an exception. This issue can occur due to several reasons:

- Incorrect disk information:

  Use the `GET /resource/pkgs` API to verify disk parameters. If the information appears incorrect, contact the environment administrator.
- Incorrect network information:

  Use the `GET /resource/pkgs` API to verify network parameters. If the information appears incorrect, contact the environment administrator.
- Missing image location:

  ACC cannot locate the image on its disk. Use the `GET /images` API to check available images.
- Appliance reboot delay:

  After installation, ACC waits for the appliance to reboot. If the appliance does not respond to ACC calls post-reboot, this exception may occur. Check the appliance status on the HMC. If the appliance is still rebooting, wait until it completes. If the appliance becomes responsive, you can ignore this exception.
- Installation failure due to HMC error:

  For example, the LPAR may not be activated. Check the HMC UI and logs for error messages.

User response:
If all configurations appear correct, gather the ACC appliance logs using the `*_pull_ssc_logs.yaml` Ansible playbook and contact IBM Support.

Descriptor code:

E01103

E01104 - Importing appliance configuration failed during upgrade	**Importing the appliance configuration has failed**

Explanation

When upgrading an appliance using ACC, ACC first gathers the required configuration files from the appliance. These files are sent to the user and stored internally.

After the upgrade completes (i.e., the appliance disk content is updated), ACC attempts to reboot the appliance and restore its configuration. If this step fails, ACC throws an exception. This issue arises due to one of the following reasons:

- Appliance is not active:

  Check the appliance status using the HMC UI or the appliance's API/UI. If the appliance is inactive and cannot be activated, use the workaround below or contact IBM Support.

- Appliance corruption after update:

  If corruption is suspected, collect logs using the `*_pull_ssc_logs.yaml` Ansible playbook and contact IBM Support. You can verify appliance health using REST APIs or the appliance's web UI.

- Reboot failure:

  If the appliance fails to respond after reboot, check its status on the HMC UI. The appliance may require additional time to complete the reboot. If it becomes responsive, this exception can be ignored.

User response

Reinstall the original version of the appliance and manually restore the configuration using the SSC API:

```
POST /appliance-configuration/import
```

Refer to "Creating an ACC Restore Checkpoint" on page 31for detailed instructions on restoring configurations from a checkpoint.

Descriptor code:
E01104

E01302 - API input not valid	**User has entered an invalid value in the request.**

Explanation

When a user sends a REST API request to ACC, ACC validates the contents of the request body. If the input is incorrect or in an unexpected format, ACC throws an exception.

The most common causes are:

- A field exceeds the allowed size.
- A string field contains an integer.
- A list field contains a string.
- Required fields are missing.
- Field values are invalid or improperly formatted.

User response:
Follow the instructions provided in the User Guide.

Descriptor code:
E01302

E01303 - Invalid request body	**User has entered incorrect request body fields.**

Explanation

If the request body of an API is malformed, then ACC throws this exception.

The most common causes are:

- Invalid values used for the fields.
- Missing a required field.

User response:
Follow the instructions provided in the User Guide.

Descriptor code:
E01303

E01304 - Invalid user role	**The role of the user is not allowed to run this task**

Explanation

Certain actions in ACC are restricted based on user roles. Some actions can only be performed by the ACC administrator, while others are reserved for appliance owners.

If an ACC administrator attempts to perform an appliance-owner action—or vice versa—ACC throws an exception.

User response

- Review the user guide to understand role-based permissions.
- If you are an appliance owner and need to perform an ACC administrator action, contact the ACC administrator directly.

Descriptor code:
E01304

E01305 - Invalid Query parameters **The user entered incorrect query parameter.**

Explanation

This exception occurs when ACC expects query parameters, but they are entered incorrectly by the user.

The most common cause are:

- The length of a query parameter exceeds the allowed limit.
- The query parameter is not a valid string.
- The query parameter contains invalid or incorrect values

User response:
Follow the instructions provided in the user guide.

Descriptor code:
E01305

E01307 - LPAR is not in desired state **LPAR is not in the right state.**

Explanation

For certain actions such as activation or deactivation, ACC expects the LPAR to be in a specific state.

Before performing the action, ACC checks the current state of the LPAR. If the state is unexpected, ACC throws an exception.

This exception can occur due to several reasons:

Locked Appliance

If ACC attempts to gather information about an appliance that is in a locked state (i.e., ACC does not have credentials to access it), the exception is triggered.

To resolve this, unlock the appliance using the unlock API or ACC's UI before proceeding.

Installer State Conflict

If ACC tries to perform an operation while the appliance is in installer state, the exception may occur. Check the appliance's state using the HMC UI.

To fix the issue, you can try debugging appliance issues, restarting the appliance (with caution) and , reviewing HMC logs for further insight.

User response:
If all configurations appear correct, gather the ACC appliance logs using the `*_pull_ssc_logs.yaml` Ansible playbook and contact IBM Support.

Descriptor code:
E01307

E01308 - API input has wrong type **The input value is not of the right type.**

Explanation:
The exception is raised when an incorrect value type is used. For example, a string is passed instead of int.

User response:
Follow the instructions provided in the user guide and enter the correct type of values.

Descriptor code:
E01308

E01310 - HMC configuration is in progress **ACC is busy configuring HMC connectivity.**

Explanation

In default mode, the ACC communicates with the HMC. The ACC must configure its connectivity with the HMC using HMC APIs. If this configuration is in progress and a request is received by the ACC that could disrupt the process, an exception is thrown.

The user should retry the command after a few minutes. For example, five minutes. If the process takes longer than expected, verify that the network connectivity between the ACC and the HMC is stable and functioning properly.

Additionally, if the HMC manages multiple CPCs, and each CPC contains numerous LPARs, the configuration may require more time. If the HMC is located in a geographically remote area, the ACC may also take longer to complete the configuration.

User response:
If all configurations appear correct, gather the ACC appliance logs using the `*_pull_ssc_logs.yaml` Ansible playbook and contact IBM Support.

Descriptor code:
E01310

E01311 - Unable to fetch CPC configuration from HMC **ACC cannot collect information of a CPC.**

Explanation

In default mode, ACC communicates with the HMC. ACC must configure its connectivity with the HMC using HMC APIs. This configuration involves gathering information about the CPCs including their resources, active LPARs etc. If this information gathering process fails, then this exception is thrown.

ACC-admin should check the status of the CPC on the HMC's UI. If the CPC is down, then contact the environment admin to fix the issue. Also, check whether the HMC admin has granted ACC-admin access to the CPC.

User response:
If all configurations appear correct, gather the ACC appliance logs using the `*_pull_ssc_logs.yaml` Ansible playbook and contact IBM Support.

Descriptor code:
E01311

E01312 - Task deletion failed **Cannot delete the task.**

Explanation

Certain commands sent to the ACC may initiate long-running tasks, such as appliance installation or upgrade. Occasionally, these tasks may become stuck due to errors and fail to progress. In such cases, the user may attempt to delete the task. If task deletion fails, this exception is thrown

There are several possible reasons for task deletion failure:

- The task is still in progress when the user invokes the task deletion API.
- The specified task ID cannot be found

User response

- Determine whether it is necessary to delete the task. If the task is not essential and does not pose any harm, consider ignoring the error.
- Check the status of the appliances associated with the task using the HMC interface. Try to resolve any issues directly on the HMC.
- If all systems are functioning as expected and task deletion is still required, restart the ACC and verify whether the task has been removed.
- If all configurations appear correct, gather the ACC appliance logs using the `*_pull_ssc_logs.yaml` Ansible playbook and contact IBM Support.

Descriptor code:
E01312

E01402 - MFA is not enabled **Action requires MFA to be enabled on ACC.**

Explanation

If a user action assumes that multi-factor authentication (MFA) is enabled on the ACC, and it is not, this exception is thrown.

For example, if the ACC administrator attempts to create a secret for appliance owners or for the ACC administrator while MFA is disabled on the ACC, the exception will occur.

To enable and use MFA, the ACC administrator should use the PUT `/mfa/enable` API.

User response:
Enable MFA or contact the ACC-admin

Descriptor code:
E01402

E01403 - MFA enabled but secret not updated **ACC cannot use MFA because the secret is not updated.**

Explanation

When the ACC administrator enables multi-factor authentication (MFA), each user receives a temporary MFA secret. This secret can be used to update the password and obtain a new MFA secret. The new MFA secret must then be used for time-based one-time password (TOTP) generation.

If the user does not retrieve the new MFA secret and attempts to use MFA, this exception is thrown. To resolve the issue, the user should update their password and use the returned MFA secret in their TOTP generator.

User response:
Update the secret before using MFA

Descriptor code:
E01403

E01404 - MFA is enabled, OTP is not provided **ACC is waiting an TOTP.**

Explanation:
If MFA is enabled and TOTP is not provided, then this exception is thrown. You must use a TOTP generator along with the secret provided by the PUT `/user` API (password update) and the generated TOTP when using the command.

User response:
Provide the OTP.

Descriptor code:
E01404

E01405 - Invalid OTP **The OTP provided for MFA is incorrect.**

Explanation

ACC supports two-factor authentication (2FA) where you must provide a TOTP to ACC for authentication. If the TOTP is invalid, then this exception is thrown. Some reasons this TOTP is invalid could be:

- When an incorrect secret is used for generating TOTP.
- When the user's password is not updated.
- TOTP is expired. The TOTP valid for 30 seconds only.

User response:
Contact the ACC-admin in case the TOTP based authentication is failing.

Descriptor code:
E01405

E01406 - MFA enabled failed due to error in encrypting and updating secret for user **The encryption of MFA secret has failed.**

Explanation:
When ACC-admin enables MFA, ACC creates a new temporary secret for all users. ACC protects the key by encrypting the key before storing it internally. If this encryption process fails, this exception is thrown.

User response

- Temporarily disable MFA.
- Gather the ACC appliance logs using the `*_pull_ssc_logs.yaml` Ansible playbook and contact IBM Support.

Descriptor code:
E01406

E01407 - MFA is already enabled **Administrator tried to enable MFA. However, the MFA already enabled.**

Explanation:
If the ACC-admin uses the PUT `/mfa/enable` when the MFA is already enabled, then this exception is thrown.

User response:
The ACC-admin should proceed with other APIs to update the password, and create temporary secrets for the users.

Descriptor code:
E01407

E01409 - MFA secret is not generated **The MFA secret does not exist for the user.**

Explanation:
When user tries to use MFA enabled ACC with an user account which does not have a secret associated with the ACC, then this exception is thrown.

User response:
ACC-admin must use the POST `/mfa/secret/owner` API.

Descriptor code:
E01409

E01501 - Duplicate resource found **Resource package name already exists.**

Explanation

To handle and manage resources, the ACC-admin must create resource packages for the appliance-owners. Each resource package has a unique "name". If resource name already exists then, this exception is thrown.

User response:
The ACC-admin should retry creating the resource package with a different name.

Descriptor code:
E01501

E01502 - Resource not found **A few of the resources are not found in ACC.**

Explanation

If you send a command that references a resource not found by the ACC, the system throws this exception.

Several scenarios may lead to this exception:

- The name or ID of the resource such as a resource package, LPAR, owner, image, task, or alert is incorrect.
- The user has entered an invalid LPAR or appliance name that does not exist within the specified resource package.
- The ACC administrator attempts to create a resource package using LPARs that the ACC cannot locate on the HMC.

User response:
The ACC-admin should retry creating the resource package with a different name.

Descriptor code:
E01502

E01503 - Insufficient Resources **There are insufficient resource to handle the request.**

Explanation

Many requests to the ACC involve reserving and consuming system resources such as logical partitions (LPARs), integrated facility for Linux (IFLs), and memory. If a user attempts to consume more resources than have been reserved, allocated, or are currently available, the ACC throws a resource consumption exception.

Common scenarios that trigger this exception:

- A user requests 16 IFLs for a specific LPAR, but the ACC administrator has only allocated 8 IFLs in total to that user.
- A user requests 4 IFLs for a new LPAR, but is already using 6 IFLs across other LPARs, while the total allocation is only 8 IFLs. In this case, only 2 IFLs are available for the new request.
- An appliance image requires at least 4 GB of memory, but the user attempts to install it on an LPAR with only 2 GB.
- A user requests an LPAR with 16 GB of memory, but the CPC has only 8 GB of memory available, even though the ACC administrator has allocated 32 GB to the user.
- A new version of an appliance image is used to upgrade an existing appliance, but the updated version requires more resources than those currently allocated.

User response:
Increase resource allotment.

Descriptor code:
E01503

E01601 - SSC API call failure **ACC received an unknown error when calling an SSC appliance API.**

Explanation

The ACC communicates with SSC appliances on behalf of appliance owners. If a request sent from the ACC to an appliance results in an error that the ACC does not recognize, this exception is thrown.

Common scenarios that trigger this exception:

- When you attempt to update an appliance, the ACC internally tries to retrieve the appliance's configuration using its API. If the API call fails, the ACC throws this exception to inform the user.
- If the ACC is configured to use multi-factor authentication (MFA) with an appliance that does not support MFA, the appliance will return an error. This error is then surfaced to the user as this exception.

User response

To resolve this issue:

- Check the status of the appliance you are trying to manage through the ACC.
- Ensure the appliance is healthy and functioning as expected.
- Review the appliance logs on the HMC for any underlying issues or error messages.
- Gather the ACC appliance logs using the `*_pull_ssc_logs.yaml` Ansible playbook and contact IBM Support.

Descriptor code:
E01601

E01602 - ACC received an error when calling an SSC appliance API **ACC received an error when calling an SSC appliance API.**

Explanation

The ACC communicates with SSC appliances on behalf of appliance owners. If a request from the ACC to an appliance results in an error response specifically an HTTP 500 status code this exception is thrown.

This typically indicates an internal error on the appliance side that the ACC cannot resolve or interpret directly.

User response

- Check the status of the appliance you are attempting to manage through the ACC.
- Ensure the appliance is healthy and functioning as expected.
- Review the appliance logs on the HMC for any error messages or system faults.
- Gather the ACC appliance logs using the `*_pull_ssc_logs.yaml` Ansible playbook and contact IBM Support.

Descriptor code:
E01602

E01603 - SSC authentication error **ACC could not authenticate itself with the SSC appliance.**

Explanation

ACC communicates with the SSC appliances on behalf of the appliance-owners. Each request from the ACC to the appliances requires authentication. If the authentication fails and ACC receives the return code 401 from the SSC appliances, then this exception is thrown.

To solve such issues perform the following:

- Check the status of the appliance that you would like to handle using ACC.
- If you have changed the credentials of the appliance, then update these credentials in the ACC using the `POST /unlock_quota` API.
- Check if the appliance is healthy and working as expected.
- Check the logs of the appliance on the HMC.

User response:
Gather the ACC appliance logs using the `*_pull_ssc_logs.yaml` Ansible playbook and contact IBM Support.

Descriptor code:
E01603

E01604 - MFA is Disabled in Appliance **MFA is disabled by the SSC appliance.**

Explanation

ACC communicates with the SSC appliances on behalf of the appliance-owners. Each request from the ACC to the appliances requires authentication. If ACC is configured to use MFA with the appliance, and the SSC appliance disables the MFA, then this exception.

User response

- Update the access credentials in the ACC using the `POST /unlock_quota API`.
- Retry the API again, unlock appliance if required.

Descriptor code:
E01604

E01606 - SSC Client Exception **ACC cannot communicate with the SSC appliance.**

Explanation

The ACC communicates with SSC appliances on behalf of appliance owners using an internal client that interfaces with the appliances using a REST API.
If this internal client encounters an error during communication, this exception is thrown.

- The SSC appliance is running an older version that does not support all required APIs.
- The SSC appliance has not implemented the specific API being requested by the ACC's internal client.
- The SSC appliance is unreachable due to network issues or downtime.
- The appliance was restarted and is no longer reachable.
- A timeout occurred while waiting for a response from the appliance.

User response

To resolve the issue:

- Check the status of the appliance using the HMC user interface.
- Review the appliance logs on the HMC for any errors or warnings.
- Verify the health of the appliance using the

  ```
  POST /cluster/health
  ```

  API to ensure it is functioning as expected.

Descriptor code:
E01606

E01607 - LPAR is not in appliance mode **SSC appliance is not in the appliance mode.**

Explanation

ACC communicates with the SSC appliances on behalf of the appliance-owners. For some of these actions, the ACC expects the SSC appliance to be in the proper state. For example, if the appliance is in the "installer" mode and some action such as health monitoring is triggered by the user, then ACC throws this exception.

User response

- Use the POST /unlock_quota API to unlock the appliance and transition it into the right appliance mode.
- If this API fails, check the status of the appliance on the HMC's UI and debug using the appliance logs.
- Gather the ACC appliance logs using the `*_pull_ssc_logs.yaml` Ansible playbook and contact IBM Support.

Descriptor code:
E01607

E02002 - Invalid state **An invalid state has been encountered by ACC.**

encountered by ACC

Explanation

For some of the actions, ACC expects resources such as LPARs to be in a specific state. If the states are incorrect, then this exception is thrown. For example:

- If the ACC is used to activate an LPAR that is already active
- If the ACC is used to deactivate an LPAR which is already in deactivate state.
- If the ACC is used to deactivate an LPAR that has an already associated activate or deactivate task running.

To fix this issue

- Check the state of the appliances on the HMC, and wait till the currently running task is finished. If required, manually sync the state of the LPARs and CPCs with the ACC using `POST  /sync` APIs.

User response:
Sync ACC with LPARs/CPCs and wait for tasks to finish.

Descriptor code:
E02002

E02005 -Database Error Occurred **ACC encountered an internal database error.**

Explanation

ACC uses a database to store information about appliances, LPARs, users etc. If accessing or modifying the database results in an error, this exception is thrown.

User response

- Wait for the ongoing task to complete before retrying your request.
- Monitor the appliance’s state using the HMC user interface to determine when the resource becomes available.
- If necessary, perform disruptive actions on the HMC to reset or recover the resource. Proceed with caution, as such actions may result in the loss of unsaved data.

Descriptor code:
E02005

E02006 -Duplicate entry found for resource **ACC encountered an internal database error.**

Explanation

When a user sends a request to the ACC for resources such as LPARs, and another request is already interacting with the same resource, this exception is thrown. This typically occurs when concurrent operations conflict for example, if a user initiates an activation or deactivation request while another ACC internal task is already performing activation or deactivation on the same LPAR.

User response

- Wait for the ongoing task to complete before retrying your request.
- Monitor the appliance’s state using the HMC user interface to determine when the resource becomes available.
- If necessary, perform disruptive actions on the HMC to reset or recover the resource. Proceed with caution, as such actions may result in the loss of unsaved data.

Descriptor code:
E02006

E02101 -LPAR is in invalid mode **LPAR is in invalid mode to perform the requested action.**

Explanation

ACC communicates with the SSC appliances on behalf of the appliance-owners. For some of these actions, the ACC expects the SSC appliance to be in the proper state. For example, if the appliance is in the "installer" mode and some action such as health monitoring is triggered by the user, then ACC throws this exception.

User response

- Use the POST /unlock_quota API to unlock the appliance and transition it into the right appliance mode.
- If this API fails, check the status of the appliance on the HMC's UI and debug using the appliance logs.
- Gather the ACC appliance logs using the `*_pull_ssc_logs.yaml` Ansible playbook and contact IBM Support.

Descriptor code:
E02101

E02102 - ACC internal error while fetching network interface information **Network information could not be fetched internally by ACC.**

Explanation

ACC stores and fetches the network information of SSC appliances. If ACC is unable to fetch the information, then ACC throws this exception.

User response:
Gather the ACC appliance logs using the `*_pull_ssc_logs.yaml` Ansible playbook and contact IBM Support.

Descriptor code:
E02102

E02104 - Failure in updating HMC Config	**ACC failed in updating the HMC configuration.**

Explanation

In default mode, the ACC communicates with the HMC. To establish this communication, the ACC must store the ACC-user credentials for the HMC. These credentials are securely stored in an encrypted memory buffer within the ACC.

This buffer is automatically flushed every 24 hours, requiring the ACC administrator to update the HMC configuration daily. If the request to update this configuration fails, the ACC throws this exception.

User response:
Gather the ACC appliance logs using the `*_pull_ssc_logs.yaml` Ansible playbook and contact IBM Support.

Descriptor code:
E02104

E02105 - Error while retrieving ACC information	**Could not retrieve ACC information.**

Explanation:
This exception is thrown when ACC fails to get information on ACC

User response:
Gather the ACC appliance logs using the `*_pull_ssc_logs.yaml` Ansible playbook and contact IBM Support.

Descriptor code:
E02105

E02201 - Error while connecting to HMC	**ACC encounter an error while making connections to HMC.**

Explanation:
In the default mode, ACC establishes the connection with the HMC using the stored credentials, and then use this connection to send requests to HMC. If establishing of this connection fails, ACC throws this exception.

User response:
Check the connectivity of the ACC with the HMC by running the `GET /cpcs/hmc-connection` API.

Descriptor code:
E02201

E02301 - Failure to retrieve alerts from appliance	**ACC is unable to retrieve alerts or logs from appliances.**

Explanation:
ACC can retrieve a list of alerts and logs from a cluster of appliances. If the SSC appliance returns a list of alerts that is incorrect or malformed, then this exception is thrown.

User response

- Check whether the SSC appliance is working properly.
- Use the latest version of SSC appliance.

Descriptor code:
E02301

E02401 - Error while trying to restart appliance	**ACC encountered an error while restarting the appliance.**

Explanation:
If the appliance-owner requests the ACC to restart an appliance and the task cannot be initiated, then this exception is thrown.

User response

- Check the status of the appliance on the HMC's UI. If the appliance restarted as expected, then consider ignoring this error
- Gather the ACC appliance logs using the `*_pull_ssc_logs.yaml` Ansible playbook and contact IBM Support.

Descriptor code:
E02401

E02502 -Error occurred while uploading the certificate	**Uploaded certificate does not match the ACC's key.**

Explanation:

If the ACC-admin uploads the CA signed certificate to ACC and if the certificate keys do not match with the key generated by ACC, then this exception is thrown.

User response

- Recreate the Certificate Signing Request (CSR) in ACC using the POST `/certificate/csr` API, get the CSR signed, and upload the CA signed certificate in the PEM format using POST `/certificate/upload` API.
- If all configurations appear correct, gather the ACC appliance logs using the `*_pull_ssc_logs.yaml` Ansible playbook and contact IBM Support.

Descriptor code:
E02502

E02503 - Error while uploading certificate **ACC's CA signed certificate has expired.**

Explanation

If the ACC-admin uploads an expired CA signed certificate to ACC, then this exception is thrown.

User response

- Recreate the Certificate Signing Request (CSR) in ACC using the POST `/certificate/csr` API, get the CSR signed, and upload the CA signed certificate in the PEM format using POST `/certificate/upload` API.
- If all configurations appear correct, gather the ACC appliance logs using the `*_pull_ssc_logs.yaml` Ansible playbook and contact IBM Support.

Descriptor code:
E02503

E02504 - Failed to reload the Nginx while uploading the certificate **Failed to reload the Nginx while uploading the certificate.**

Explanation:
Some processes within ACC require reloading of the Nginx configuration. If reloading fails, then this exception is thrown.

User response:
Gather the ACC appliance logs using the `*_pull_ssc_logs.yaml` Ansible playbook and contact IBM Support.

Descriptor code:
E02504

E02505 - Uploading certificates to ACC has failed **Storing the certificate within ACC has failed.**

Explanation:
When the ACC-admin uploads the CA signed certificate to ACC, ACC will store that certificate to a certain location. If the upload fails, then this exception is thrown. For example, the disk path within ACC becomes unreachable, then this exception will be thrown.

User response:
Gather the ACC appliance logs using the `*_pull_ssc_logs.yaml` Ansible playbook and contact IBM Support.

Descriptor code:
E02505

E02506 - Failed to clean old certificate folders **Failed to clean old certificate folders**

Explanation:
When ACC fails to perform a cleanup operation of old certificates, then this exception occurs. For example, if ACC is unable to reach the disk or path that contains the old certificates, this exception will be thrown.

User response:
Gather the ACC appliance logs using the `*_pull_ssc_logs.yaml` Ansible playbook and contact IBM Support.

Descriptor code:
E02506

E02507 - Error while generating CSR certificate **ACC has failed to generate a key.**

Explanation:
ACC will internally create keys for the Certificate Signing Request (CSR). If this process fails, this exception is thrown.

User response:
Gather the ACC appliance logs using the `*_pull_ssc_logs.yaml` Ansible playbook and contact IBM Support.

Descriptor code:
E02507

E02508 - Certificate operation already in progress **ACC is currently busy in a certification operation.**

Explanation:
Whenever a request comes to ACC for managing certificates and if ACC is already busy with another operation that involves certificates, then this exception is thrown.

User response:
Wait until the other certificate operation are finished

Descriptor code:
E02508

Notices

This information was developed for products and services offered in the US.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive, MD-NC119
Armonk, NY 10504-1785
US

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some jurisdictions do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM websites are provided for convenience only and do not in any manner serve as an endorsement of those websites. The materials at those websites are not part of the materials for this IBM product and use of those websites is at your own risk.

IBM may use or distribute any of the information you provide in any way it believes appropriate without incurring any obligation to you.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

Statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to actual people or business enterprise is entirely coincidental.

Trademarks

IBM, the IBM logo, and ibm.com® are trademarks or registered trademarks of International Business Machines Corporation, in the United States and/or other countries. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on http://www.ibm.com/trademark.

Class A Notices

The following Class A statements apply to this IBM product. The statement for other IBM products intended for use with this product will appear in their accompanying manuals.

Federal Communications Commission (FCC) Statement

Note: This equipment has been tested and found to comply with the limits for a Class A digital device, pursuant to Part 15 of the FCC Rules. These limits are designed to provide reasonable protection against harmful interference when the equipment is operated in a commercial environment. This equipment generates, uses, and can radiate radio frequency energy and, if not installed and used in accordance with the instruction manual, may cause harmful interference to radio communications. Operation of this equipment in a residential area is likely to cause harmful interference, in which case the user will be required to correct the interference at his own expense.

Properly shielded and grounded cables and connectors must be used in order to meet FCC emission limits. IBM is not responsible for any radio or television interference caused by using other than recommended cables and connectors or by unauthorized changes or modifications to this equipment. Unauthorized changes or modifications could void the user's authority to operate the equipment.

This device complies with Part 15 of the FCC rules. Operation is subject to the following two conditions: (1) this device may not cause harmful interference, and (2) this device must accept any interference received, including interference that may cause undesired operation.

Industry Canada Compliance Statement

This Class A digital apparatus complies with Canadian ICES-003.

United Kingdom Notice

This product may cause interference if used in residential areas. Such use must be avoided unless the user takes special measures to reduce electromagnetic emissions to prevent interference to the reception of radio and television broadcasts.

Avis de conformité à la réglementation d'Industrie Canada

Cet appareil numérique de la classe A est conforme à la norme NMB-003 du Canada.

European Community Compliance Statement

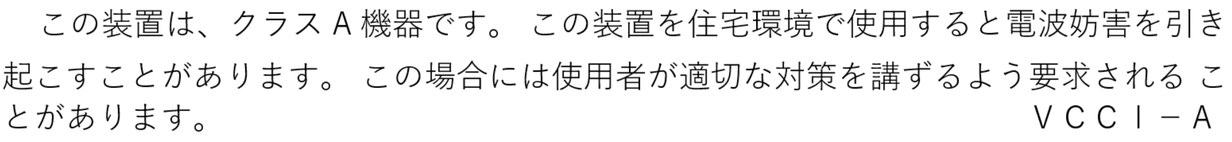
This product is in conformity with the protection requirements of EU Council Directive 2014/30/EU on the approximation of the laws of the Member States relating to electromagnetic compatibility. IBM cannot accept responsibility for any failure to satisfy the protection requirements resulting from a non-recommended modification of the product, including the fitting of non-IBM option cards.

This product has been tested and found to comply with the limits for Class A Information Technology Equipment according to European Standard EN 55032. The limits for Class A equipment were derived for commercial and industrial environments to provide reasonable protection against interference with licensed communication equipment.

European Community contact:
IBM Deutschland GmbH
Technical Regulations, Department M372
IBM-Allee 1, 71139 Ehningen, Germany
Tele: +49 (0) 800 225 5423 or +49 (0) 180 331 3233
email: halloibm@de.ibm.com

Warning: This is a Class A product. In a domestic environment, this product may cause radio interference, in which case the user may be required to take adequate measures.

Japan Voluntary Control Council for Interference (VCCI) Notice



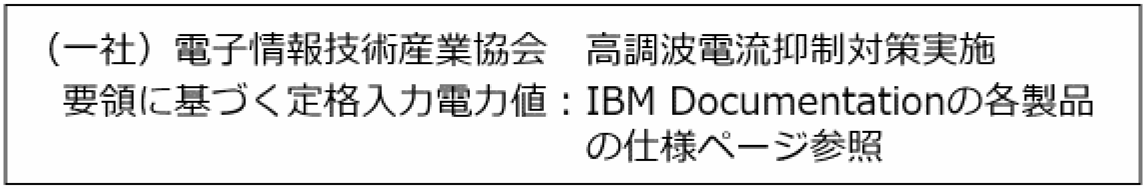
この装置は、クラス A 機器です。 この装置を住宅環境で使用すると電波妨害を引き起こすことがあります。 この場合には使用者が適切な対策を講ずるよう要求されることがあります。

ＶＣＣＩ－Ａ

The following is a summary of the Japanese VCCI statement above:

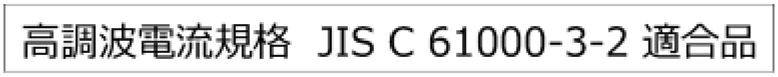
This is a Class A product based on the standard of the VCCI Council. If this equipment is used in a domestic environment, radio interference may occur, in which case the user may be required to take corrective actions.

Japan Electronics and Information Technology Industries Association (JEITA) Notice



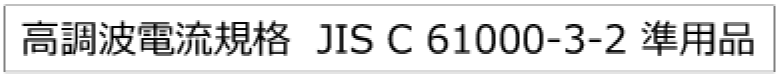
（一社）電子情報技術産業協会　高調波電流抑制対策実施
要領に基づく定格入力電力値：IBM Documentationの各製品の仕様ページ参照

This statement applies to products less than or equal to 20 A per phase.

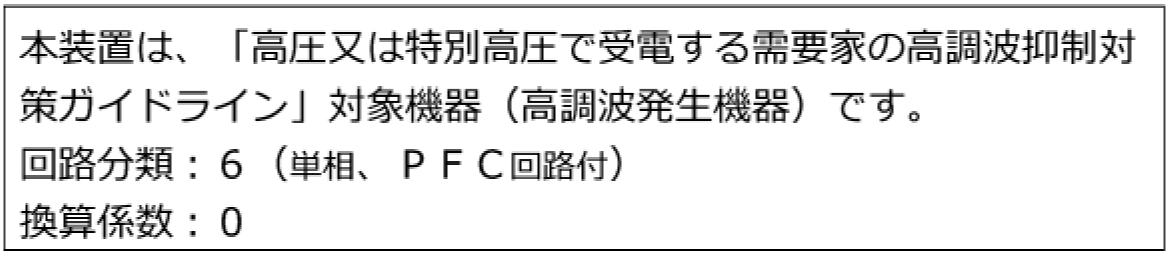


高調波電流規格 JIS C 61000-3-2 適合品

These statements apply to products greater than 20 A, single-phase.



高調波電流規格 JIS C 61000-3-2 準用品

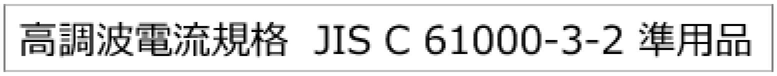


本装置は、「高圧又は特別高圧で受電する需要家の高調波抑制対策ガイドライン」対象機器（高調波発生機器）です。

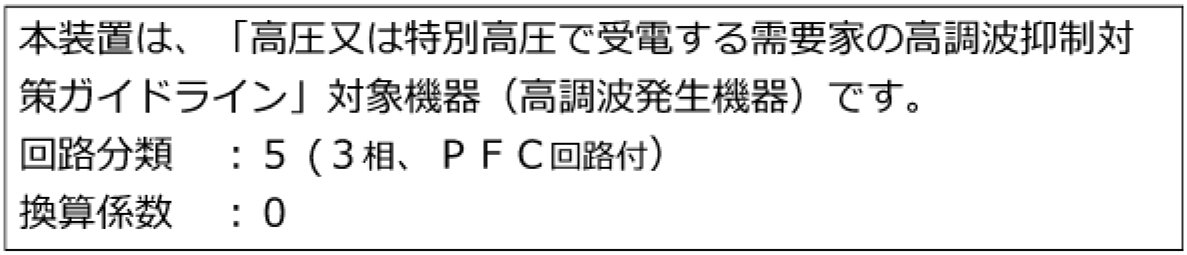
回路分類：６（単相、ＰＦＣ回路付）

換算係数：０

These statements apply to products greater than 20 A per phase, three-phase.



高調波電流規格 JIS C 61000-3-2 準用品

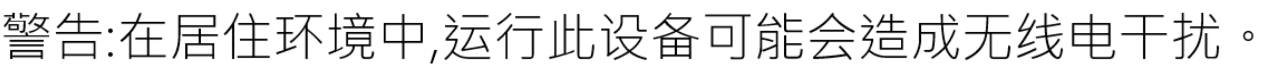


本装置は、「高圧又は特別高圧で受電する需要家の高調波抑制対策ガイドライン」対象機器（高調波発生機器）です。

回路分類　：５（３相、ＰＦＣ回路付）

換算係数　：０

People's Republic of China Notice



警告:在居住环境中,运行此设备可能会造成无线电干扰。

Declaration: This is a Class A product. In a domestic environment, this product may cause radio interference, in which case the user may need to perform practical action.

Taiwan Notice

CNS 13438:

警告使用者：
此為甲類資訊技術設備，
於居住環境中使用時，
可能會造成射頻擾動，在此種情況下，
使用者會被要求採取某些適當的對策。

CNS 15936:

警告：為避免電磁干擾，本產品不應安裝或使用於住宅環境。

IBM Taiwan Contact Information:

台灣IBM 產品服務聯絡方式：
台灣國際商業機器股份有限公司
台北市松仁路7號3樓
電話：0800-016-888

Electromagnetic Interference (EMI) Statement - Korea

이 기기는 업무용(A급)으로 전자파적합등록을 한 기기이오니
판매자 또는 사용자는 이 점을 주의하시기 바라며, 가정외의
지역에서 사용하는 것을 목적으로 합니다.

Germany Compliance Statement

Deutschsprachiger EU Hinweis: Hinweis für Geräte der Klasse A EU-Richtlinie zur Elektromagnetischen Verträglichkeit

Dieses Produkt entspricht den Schutzanforderungen der EU-Richtlinie 2014/30/EU zur Angleichung der Rechtsvorschriften über die elektromagnetische Verträglichkeit in den EU-Mitgliedsstaaten und hält die Grenzwerte der EN 55032 Klasse A ein.

Um dieses sicherzustellen, sind die Geräte wie in den Handbüchern beschrieben zu installieren und zu betreiben. Des Weiteren dürfen auch nur von der IBM empfohlene Kabel angeschlossen werden. IBM übernimmt keine Verantwortung für die Einhaltung der Schutzanforderungen, wenn das Produkt ohne Zustimmung von IBM verändert bzw. wenn Erweiterungskomponenten von Fremdherstellern ohne Empfehlung von IBM gesteckt/eingebaut werden.

EN 55032 Klasse A Geräte müssen mit folgendem Warnhinweis versehen werden:

"Warnung: Dieses ist eine Einrichtung der Klasse A. Diese Einrichtung kann im Wohnbereich Funk-Störungen verursachen; in diesem Fall kann vom Betreiber verlangt werden, angemessene Maßnahmen zu ergreifen und dafür aufzukommen."

Deutschland: Einhaltung des Gesetzes über die elektromagnetische Verträglichkeit von Geräten

Dieses Produkt entspricht dem "Gesetz über die elektromagnetische Verträglichkeit von Geräten (EMVG)". Dies ist die Umsetzung der EU-Richtlinie 2014/30/EU in der Bundesrepublik Deutschland.

Zulassungsbescheinigung laut dem Deutschen Gesetz über die elektromagnetische Verträglichkeit von Geräten (EMVG) (bzw. der EMC EG Richtlinie 2014/30/EU) für Geräte der Klasse A

Dieses Gerät ist berechtigt, in Übereinstimmung mit dem Deutschen EMVG das EG-Konformitätszeichen - CE - zu führen.

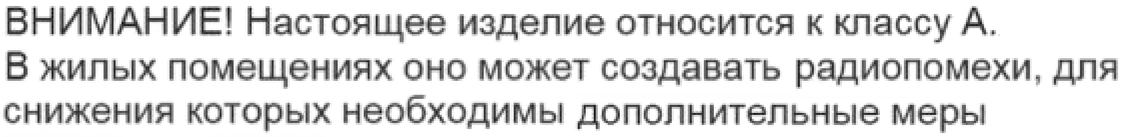
Verantwortlich für die Einhaltung der EMV Vorschriften ist der Hersteller:
International Business Machines Corp.
New Orchard Road
Armonk, New York 10504
Tel: 914-499-1900

Der verantwortliche Ansprechpartner des Herstellers in der EU ist:
IBM Deutschland GmbH
Technical Regulations, Abteilung M372
IBM-Allee 1, 71139 Ehningen, Germany
Tel: +49 (0) 800 225 5423 or +49 (0) 180 331 3233
email: halloibm@de.ibm.com

Generelle Informationen:

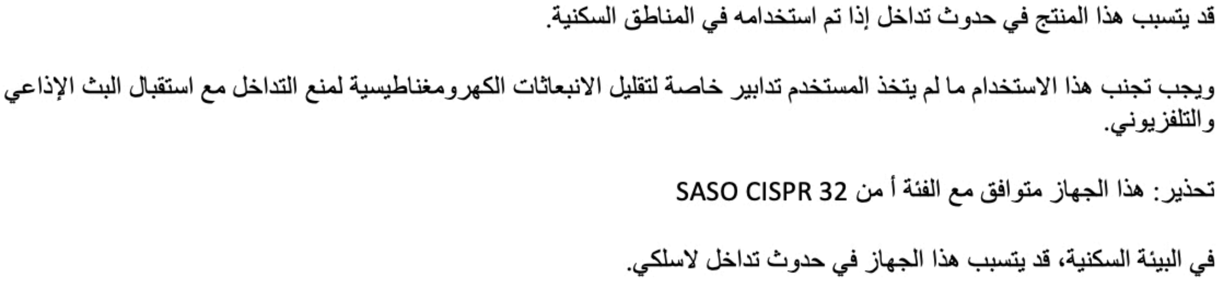
Das Gerät erfüllt die Schutzanforderungen nach EN 55024 und EN 55032 Klasse A.

Electromagnetic Interference (EMI) Statement - Russia



ВНИМАНИЕ! Настоящее изделие относится к классу А.
В жилых помещениях оно может создавать радиопомехи, для снижения которых необходимы дополнительные меры

Electromagnetic Interference (EMI) Statement - Kingdom of Saudi Arabia Notice



قد يتسبب هذا المنتج في حدوث تداخل إذا تم استخدامه في المناطق السكنية.

ويجب تجنب هذا الاستخدام ما لم يتخذ المستخدم تدابير خاصة لتقليل الانبعاثات الكهرومغناطيسية لمنع التداخل مع استقبال البث الإذاعي والتلفزيوني.

تحذير: هذا الجهاز متوافق مع الفئة أ من SASO CISPR 32

في البيئة السكنية، قد يتسبب هذا الجهاز في حدوث تداخل لاسلكي.

GC28-7073-00