



**EDB™ Ark**  
**Administrative User's Guide**

**Version 2.0**

September 15, 2016

# EDB™ Ark Administrative User's Guide

EDB™ Ark Administrative User's Guide, Version 2.0  
by EnterpriseDB Corporation  
Copyright © 2016 EnterpriseDB Corporation. All rights reserved.

EnterpriseDB Corporation, 34 Crosby Drive Suite 100, Bedford, MA 01730, USA  
**T** +1 781 357 3390 **F** +1 978 589 5701 **E** [info@enterprisedb.com](mailto:info@enterprisedb.com) **www**.[enterprisedb.com](http://enterprisedb.com)

# Table of Contents

1	Introduction.....	5
1.1	What's New .....	7
1.2	Typographical Conventions Used in this Guide.....	7
2	EDB Ark - Overview .....	8
2.1	Architecture Overview.....	8
2.2	Using OpenStack with EDB Ark .....	11
2.3	Supported Platforms.....	13
2.4	Pre-Requisites .....	13
2.4.1	Creating the EDB Ark Service Account .....	14
3	Installing EDB Ark .....	15
3.1	Step One - Importing the EDB Ark Image .....	15
3.2	Step Two – Creating the EDB Ark Security Group.....	19
3.3	Step Three - Launching the EDB Ark Console Instance .....	21
3.4	Step Four – Assign a Floating IP Address .....	23
3.5	Step Five – Configuring the Installation.....	24
3.5.1	The ppcd.properties File .....	25
3.6	Step Six - Deploying the Console.....	28
3.7	Step Seven – Configuring a User to Log In.....	29
4	Administrative Features of the EDB Ark Console .....	32
4.1	Using the Admin Tab.....	34
4.1.1	Managing Server Images .....	36
4.1.2	Managing Database Engines.....	39
4.1.3	Console Administration .....	45
4.2	Using the DBA Tab .....	49
4.3	Reference - the DBA Tables.....	51
4.3.1	activation.....	51
4.3.2	attachedvolume .....	51
4.3.3	backups .....	52
4.3.4	dbengine.....	52
4.3.5	instances.....	53
4.3.6	nodestatistics.....	55
4.3.7	pcshistory .....	55

4.3.8	pgpoolprop .....	55
4.3.9	postgresconfparameters.....	56
4.3.10	postgresprops .....	56
4.3.11	property .....	56
4.3.12	serverimage .....	57
4.3.13	snapshots .....	57
5	Securing EDB Ark .....	58
5.1	Opening a Port for SSH Connections .....	59
5.2	Using iptables Rules .....	61
6	Console Management.....	62
6.1	Starting, Stopping or Restarting the Server .....	62
6.2	Upgrading the Console .....	63
6.3	Customizing the Console .....	65
6.4	Changing Console Passwords.....	67
7	Recovering From a Console Failure .....	71
7.1	Enabling Console Backups with the EDB Ark Script .....	71
7.2	Recovering the Console .....	72
8	Notifications.....	73
9	Resources .....	77
9.1	Licenses.....	77

# 1 Introduction

EDB Ark automatically provisions EDB Postgres Advanced Server or PostgreSQL databases in single instances, high-availability clusters, or application development sandboxes in a private cloud. EDB Ark allows service providers and organizations to offer elastic and highly scalable database-as-a-service (DBaaS) environments while freeing DBAs and application developers from the rigors of setting up and administering modern and robust database environments.

In minutes, EDB Ark configures a cluster of database machines with:

- Streaming replication
- Connection pooling
- Load balancing
- Automatic failover (transaction or recovery time preferred)
- Secure data encryption
- Rotating user-scheduled backups
- Point-in-time recovery
- Elastic storage
- Elastic scale out

EDB Ark's automatic scaling of storage resources and scale out of read replicas when a database cluster reaches user-defined thresholds is especially worth noting - this functionality provides unattended, around-the-clock responsiveness to unpredictable load demands on your database infrastructure.

This document will demonstrate how to use EDB Ark in your cloud-based database management activities:

- **EDB Ark - Overview** – Section [2](#) provides information about EDB Ark functionality and architecture.
- **Installing and configuring EDB Ark** – Section [3](#) walks you through the process of installing and configuring EDB Ark.
- **Administrative Features of the EDB Ark Console** – Section [4](#) introduces you to the features that are exclusive to the EDB Ark administrator's console.
- **Securing a Cluster** - Section [5](#) walks you through how to secure an EDB Ark cluster and opening a port for SSH connections.
- **Console Management** - Section [6](#) describes how to control the Ark console manager and customize the user console.

- **Recovering from a Console Failure** - Section 7 describes how to recover from a console failure.
- **Notifications** – Section 8 describes the user notifications that will keep you informed about any changes to your EDB Ark environment.
- **Resources** – Section 9 provides a list of EnterpriseDB resources that are available if you have unanswered questions.

This document provides an introduction to EDB Ark, and is written to acquaint you with the process of configuring and using the product's core features; it is not a comprehensive guide to using Postgres database products. Depending on your operating environment (public cloud, private cloud, or traditional hardware deployment) and hosting vendor, there may be differences in EDB Ark features and functions.

For more information about using EDB Postgres products, please visit the EnterpriseDB website at:

<http://www.enterprisedb.com/documentation>

This document uses *Postgres* to mean either the PostgreSQL or EDB Postgres Advanced Server database.

## 1.1 What's New

The following features have been added to EDB Ark for release 2.0:

- The Ark console can manage clusters that reside in an OpenStack environment.
- The Ark console enables yum-based provisioning of RPM packages across all nodes of a cluster.
- EDB Ark supports a JSON-compatible API. For information about the support offered by the API, please see the *EDB Ark Getting Started Guide*.

## 1.2 Typographical Conventions Used in this Guide

Certain typographical conventions are used in this manual to clarify the meaning and usage of various commands, statements, programs, examples, etc. This section provides a summary of these conventions.

In the following descriptions a *term* refers to any word or group of words that are language keywords, user-supplied values, literals, etc. A term's exact meaning depends upon the context in which it is used.

- *Italic font* introduces a new term, typically, in the sentence that defines it for the first time.
- Fixed-width (mono-spaced) font is used for terms that must be given literally such as SQL commands, specific table and column names used in the examples, programming language keywords, etc. For example, `SELECT * FROM emp;`
- *Italic fixed-width font* is used for terms for which the user must substitute values in actual usage. For example, `DELETE FROM table_name;`
- A vertical pipe | denotes a choice between the terms on either side of the pipe. A vertical pipe is used to separate two or more alternative terms within square brackets (optional choices) or braces (one mandatory choice).
- Square brackets [ ] denote that one or none of the enclosed term(s) may be substituted. For example, [ a | b ], means choose one of “a” or “b” or neither of the two.
- Braces {} denote that exactly one of the enclosed alternatives must be specified. For example, { a | b }, means exactly one of “a” or “b” must be specified.
- Ellipses ... denote that the proceeding term may be repeated. For example, [ a | b ] ... means that you may have the sequence, “b a a b a”.

## 2 EDB Ark - Overview

EDB Ark simplifies the process of provisioning robust Postgres deployments, while taking advantage of the benefits of cloud computing. When used with EDB Postgres Advanced Server, EDB Ark also provides an Oracle-compatible DBaaS, offering dramatic cost savings and competitive advantages.

### 2.1 Architecture Overview

The Ark console and API are designed to help you easily create and manage high-availability database clusters.

Traditionally, the expression *cluster* refers to a single instance of Postgres managing multiple databases; an EDB Ark *database server cluster* is a collection of high-availability Postgres server instances that reside in a cloud or on a traditional network.

When you create a new cluster (a group of replicated database servers), EDB Ark initializes one or more Postgres instances (virtual machines) according to your specifications. EDB Ark uses Postgres streaming replication to synchronize replicas in the cluster, and pgbpool-II to implement load balancing and connection pooling among all active instances. Figure 2.1 provides a general overview of the EDB Ark architecture.

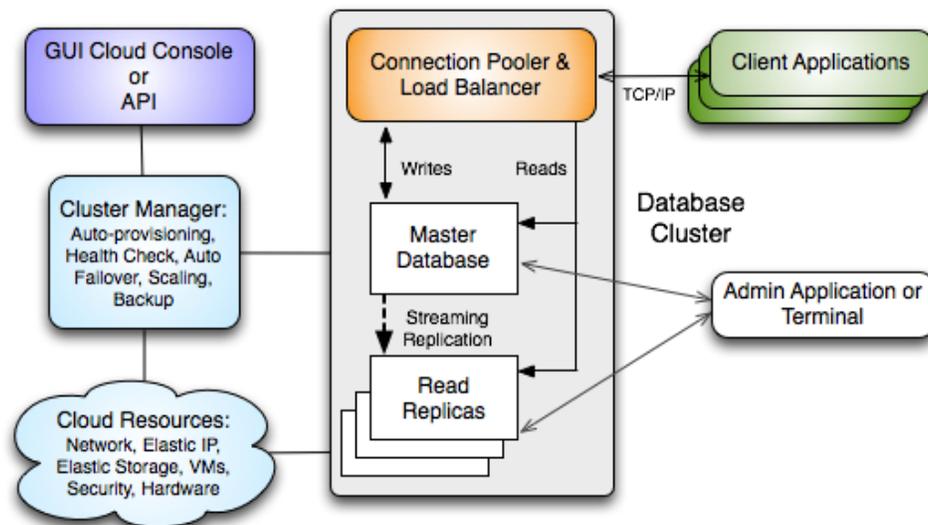


Figure 2.1 - An overview of the EDB Ark architecture.

The master node of the cluster contains a host operating system with a running instance of Postgres, along with the load balancer. Database modifications are automatically routed to the master node; any modifications to the master node are subsequently propagated to each replica using Postgres streaming replication.



instance. Automatic backups are retained according to your specifications; on-demand backups are retained until you delete them. Each backup is a complete copy of the cluster; you can use a backup to restore a cluster.

EDB Ark makes it easy to *scale* a database cluster:

- To increase read performance, you can add read replicas to the cluster (manually or automatically).
- To handle expanding data requirements you can increase the amount of storage available (manually or automatically).
- To increase the RAM or CPU processing power of the cluster's underlying virtual machine, you can manually scale a cluster into a more appropriate server class.

## 2.2 Using OpenStack with EDB Ark

A cloud (shown in Figure 2.3) is a collection of virtual machines; each virtual machine runs a separate copy of an operating system and an installation of Postgres.

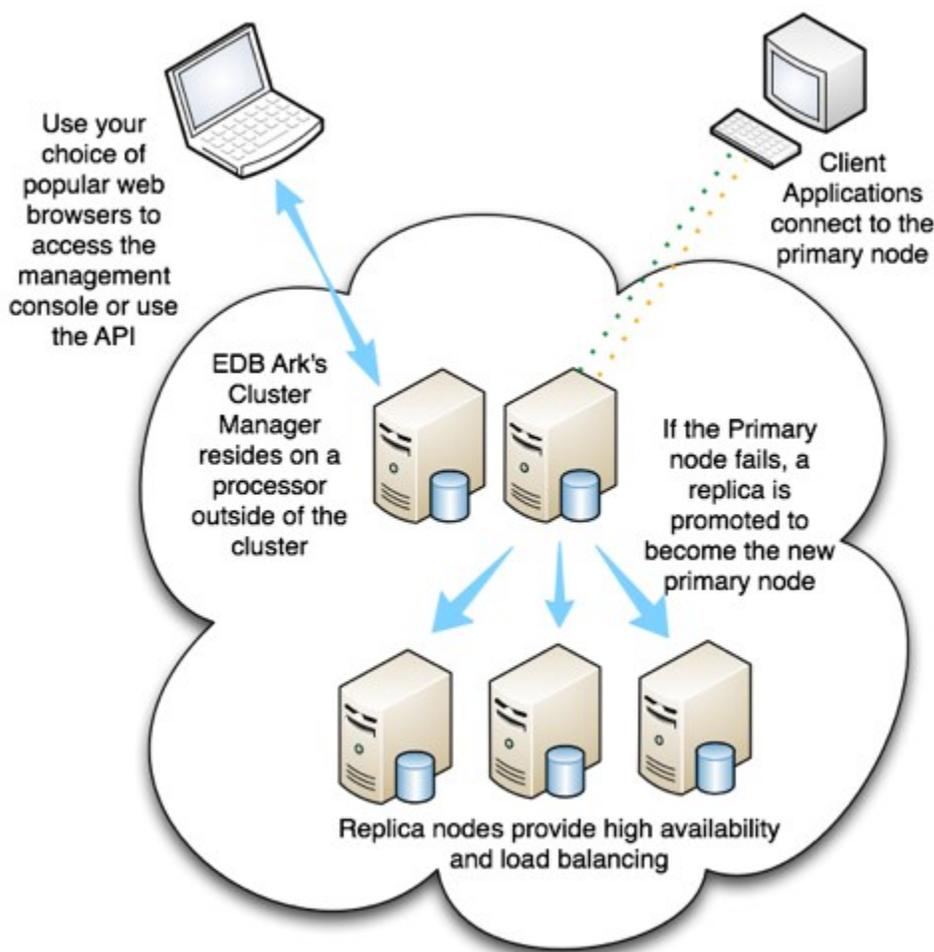


Figure 2.3 - Using EDB Ark in a Cloud.

You can specify different combinations of CPU speed, RAM, and disk space to suit your needs when provisioning an EDB Ark cluster.

When using OpenStack as a cloud provider, an OpenStack image must be registered for use as an EDB Ark *server image*. Each EDB Ark server image specifies the image ID of an OpenStack image and the name of the `default_user` that is specified in the `/etc/cloud/cloud.cfg` file associated with that image. You must register the OpenStack image in the EDB Ark Administrator's console before using it to create an EDB Ark database engine definition.

After describing the server image in the EDB Ark Administrator's console, an administrator can use the server image to define an EDB Ark *database engine*. A database engine is a combination of an OpenStack virtual machine and a database type.

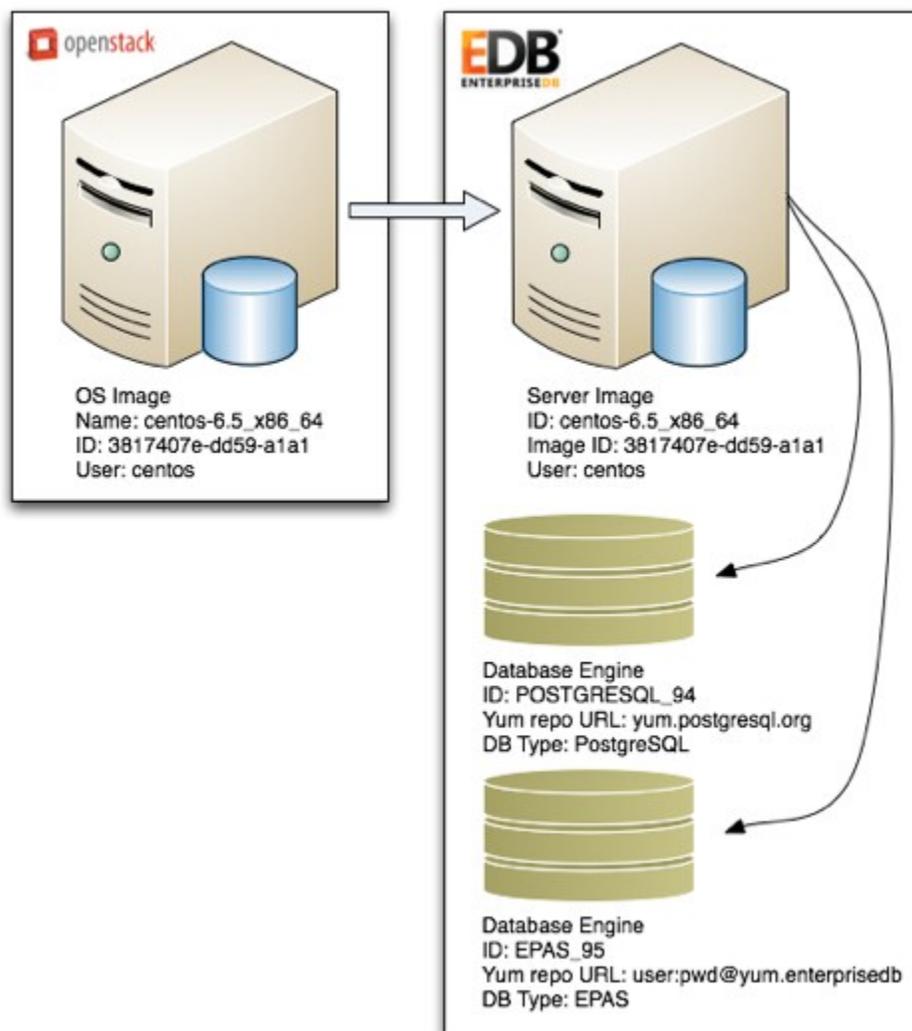


Figure 2.4 – Using an OpenStack image with EDB Ark.

The Administrator can use the same server image to create multiple database engine definitions. (see Figure 2.4). For example, you can create database engines for both PostgreSQL 9.4 and EDB Postgres Advanced Server 9.5 that are both provisioned on the same underlying server image CentOS 6.5 x86\_64.

When a user defines a cluster, the Ark console uses the information in the EDB Ark server image to launch a virtual machine (specified by the OpenStack image) to host the database server. The end-user selects the cluster configuration (the DB Engine type, size, speed and scaling preferences) in the EDB Ark end-user console.

## 2.3 Supported Platforms

EDB Ark 1.0 supports the following OpenStack versions:

Red Hat OpenStack:

- Kilo

Community OpenStack:

- Icehouse
- Kilo

EDB Ark supports provisioning on the following 64-bit Linux systems:

- CentOS 6.x and 7.x
- Red Hat Enterprise Linux 6.x and 7.x

## 2.4 Pre-Requisites

### Managing OpenStack Resource Limits

Each time the Ark console creates a cluster, a volume is created in the OpenStack management console. Each volume will have a corresponding security group, security group rules, and (if applicable) volume snapshots.

Before using the Ark console, you should ensure that OpenStack resource limits are set to values high enough to meet the requirements of your end-users. If users attempt to exceed the resource limit, the console will display an error, prompting you to increase the resource limits (see Figure 2.5).

**Notice:** {"overLimit": {"message": "SnapshotLimitExceeded: Maximum number of snapshots allowed (10) exceeded", "code": 413}}

*Figure 2.5 – A resource limit error.*

Over-restrictive limits on the following OpenStack resources may result in an error:

- volumes
- volume snapshots
- security groups
- security group rules

If a user encounters an `overLimit` error, you should connect to the OpenStack management console and increase resource limits to meet user requirements.

When you terminate a cluster that has no backups (through the Ark console), the OpenStack management console will terminate the corresponding volume and free the associated resources. If a backup of the cluster exists, the volume will persist until you delete the backup. Deleting backups of obsolete clusters will free up system resources for use.

### Restricting Access to `/var/ppcd/.edb` and the Console Properties File

The `/var/ppcd/ppcd.properties` file and the `/var/ppcd/.edb` directory contain sensitive information (including plain-text connection information) that should be accessed only by the OpenStack Administrative user. You should restrict access to the `/var/ppcd/ppcd.properties` file and the `/var/ppcd/.edb` directory, ensuring that only trusted individuals have access.

By default, the `ppcd` user has `read`, `write` and `execute` privileges on the directory (`0700`), while `group` and `other` users cannot access the directory.

## 2.4.1 Creating the EDB Ark Service Account

You must create a dedicated OpenStack user account for use by the EDB Ark service. EDB Ark uses the service account when performing OpenStack management functions. The service account user must be a member of and be assigned the OpenStack `admin` role (which is created during OpenStack installation) for all tenants that are allowed to run EDB Ark clusters.

For more information about creating an OpenStack administrative user, please consult your version-specific OpenStack documentation.

When configuring EDB Ark, you must specify the name of the OpenStack administrative role, the EDB Ark service account user name, and the password associated with the service account in the `ppcd.properties` file.

Please note that all OpenStack users that are assigned the OpenStack `admin` role will also have access to EDB Ark administrative features. Administrative users are able to register server images and create database engines, as well as retrieve information about system resources and users. For more information about the administrative features of the Ark console, see Section 4.

## 3 Installing EDB Ark

The installation instructions that follow describe the installation process on Red Hat Enterprise Linux OpenStack. OpenStack Administrative privileges are required during the installation process:

- You must be an OpenStack administrative user with sufficient privileges to upload a public image to import the EDB Ark image.
- When creating a security group and launching EDB Ark, you must use an OpenStack account with sufficient privileges in the tenant that will host the Ark console.

### 3.1 Step One - Importing the EDB Ark Image

You can use either the OpenStack dashboard GUI or the OpenStack Glance command line to import the EDB Ark image.

#### *Using the OpenStack Dashboard to Import the EDB Ark Image*

Use the following steps in the OpenStack Dashboard to import the EDB Ark image:

1. Log into the OpenStack dashboard as an administrative user.
2. Navigate to the `Admin` menu, and then select the `Images` menu selection.
3. Click the `+ Create Image` button to open the `Create An Image` dialog (shown in Figure 3.1).

**Create An Image** [X]

**Name \***

**Description**

**Description:**  
 Specify an image to upload to the Image Service.  
 Currently only images available via an HTTP URL are supported. The image location must be accessible to the Image Service. Compressed image binaries are supported (.zip and .tar.gz.)  
**Please note:** The Image Location field MUST be a valid and direct URL to the image binary. URLs that redirect or serve error pages will result in unusable images.

**Image Source**

**Image Location ⓘ**

**Format \***

**Architecture**

**Minimum Disk (GB) ⓘ**

**Minimum RAM (MB) ⓘ**

Copy Data ⓘ  
 Public  
 Protected

Cancel Create Image

*Figure 3.1 – The Create an Image dialog.*

Use fields on the Create An Image dialog to define the EDB Ark image:

- Use the **Name** field to provide a name for the image.
- Use the **Description** field to provide a description of the image.
- Use the **Image Source** drop-down listbox to specify that the source will be an Image File.
- Use the **Image Location** field to specify the location of the EDB Ark image file on your computer
- Use the **Format** drop-down listbox to select QCOW2 - QEMU Emulator.
- Enter `x86_64` in the **Architecture** field.
- Enter `16` in the **Minimum Disk (GB)** field.

- Enter 4096 in the `Min RAM (MB)` field.
- Check the box next to `Copy Data`.
- Check the box next to `Public`.
- Check the box next to `Protected`.

After completing the dialog, click the `Create Image` button to create the EDB Ark image. Please note that the process of creating an image may take a while depending on your network conditions. While the image is being created you should not exit the OpenStack dashboard or close your browser tab as it will stop the file transfer.

### ***Using the Glance Command Line to Import the EDB Ark Image***

You can also use the Glance command line tool to import the EDB Ark image. Please consult your platform-specific documentation for Glance installation instructions. After installing Glance, connect to the server as an administrator, and invoke the following command:

```
glance \
  --os-username administrative_user \
  --os-password password \
  --os-tenant-name tenant_name \
  --os-auth-url http://identity_service_name:35357/v2.0
image-create \
  --name 'image_name' \
  --disk-format qcow2 \
  --container-format bare \
  --is-public True \
  --is-protected True \
  --min-disk 16 \
  --min-ram 4096 \
  --property 'description=image_details' \
  --progress \
  --property os_type=linux
/path_to_image_file
```

#### **Where:**

*administrative\_user* is the name of an OpenStack administrative user with sufficient privileges to import the image.

*password* is the password associated with the administrative user account.

*tenant\_name* is the name of a tenant that the `--os-username` belongs to; it will be used as part of the OpenStack authentication process.

*identity\_service\_name* is the URL of the node hosting the OpenStack keystone authentication service. When importing an image, you should specify port 35357 to ensure that the required operations are available.

*image\_name* is a descriptive name of the EDB Ark image.

*image\_details* is a user-friendly description of the EDB Ark image that you are importing. For example, you might want to specify that you are importing: EDB Ark 2.0 Console on CentOS 6.6 x86\_64 Default user: centos

*path\_to\_image\_file* specifies the location and file name of the EDB Ark image file.

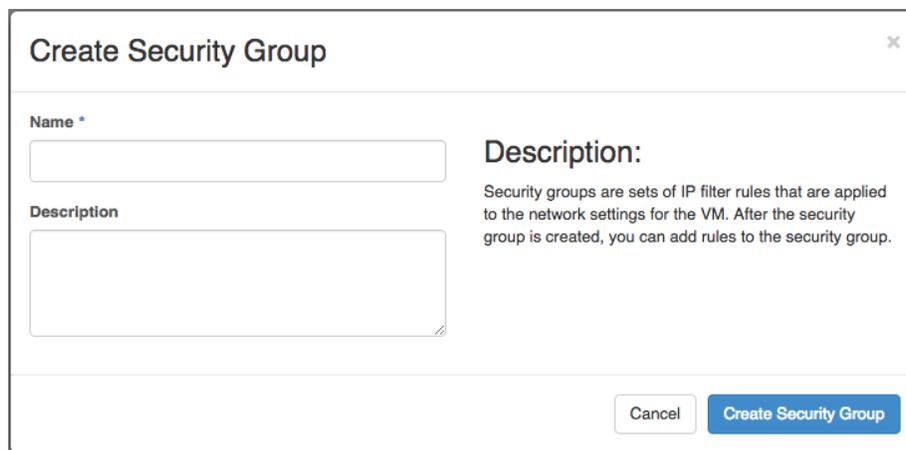
For more information about the other options supported by Glance, please consult the Glance documentation, available at:

<http://docs.openstack.org/developer/glance/>

### 3.2 Step Two – Creating the EDB Ark Security Group

The security group for the Ark console must allow communications between the nodes of the cluster. To define the security group rules:

1. Log into the OpenStack dashboard as an administrator
2. Navigate into the tenant that is hosting the Ark console.
3. Navigate to the `Project` page, and select `Access & Security`.
4. Select the `Security Group` tab, and click the `+ Create Security Group` button to open the `Create Security Group` dialog (shown in Figure 3.2).



**Create Security Group** ✕

**Name \***

**Description**

**Description:**

Security groups are sets of IP filter rules that are applied to the network settings for the VM. After the security group is created, you can add rules to the security group.

*Figure 3.2 – The Create Security Group dialog.*

Use fields on the dialog to create a security group for the image:

- Use the `Name` field to provide a name for the security group.
- Use the `Description` field to provide a description of the security group.

Click the `Create Security Group` button to create the security group and continue.

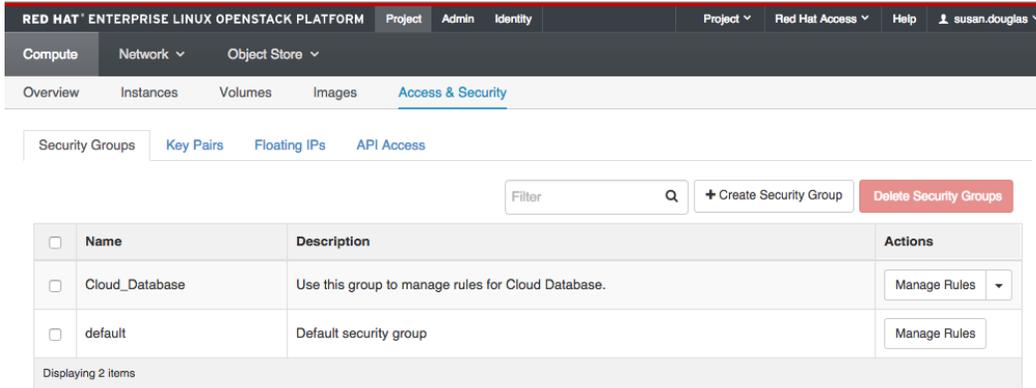


Figure 3.3 – The new group, displayed in the Security Groups list.

To add rules to the new security group, click the `Manage Rules` button that is located to the right of the security group name (see Figure 3.3). When the list of security group rules opens (see Figure 3.4), click the `+ Add Rule` button to access a dialog that allows you to add a new rule.

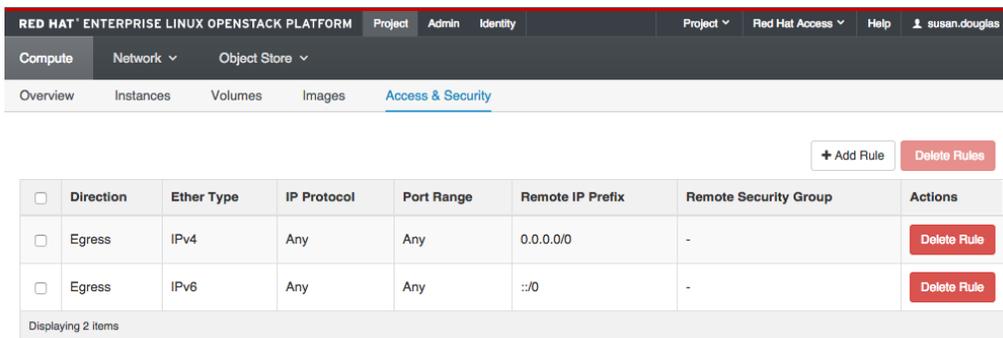


Figure 3.4 – The Security Group rules for the new security group.

Before using EDB Ark, you should add rules that allow communication between the nodes of the cluster. Use the `Add Rule` dialog to define the rules listed below:

Rule Type	Direction	Port	Remote	CIDR Address
All ICMP	Ingress		CIDR	0.0.0.0/0
SSH			CIDR	0.0.0.0/0
HTTP			CIDR	0.0.0.0/0
HTTPS			CIDR	0.0.0.0/0
Custom TCP	Ingress	6666	CIDR	0.0.0.0/0
Custom TCP	Ingress	port range from 7800 to 7900	CIDR	0.0.0.0/0
Custom TCP	Ingress	5432	CIDR	0.0.0.0/0

Please note that the last rule listed is optional, and allows access to the Postgres database that is running on the console node.

### 3.3 Step Three - Launching the EDB Ark Console Instance

After importing the image and defining the security group, you are ready to launch the Ark console instance. Log into the Openstack dashboard as an Administrative user, and navigate into the tenant that is hosting the installation. Then, navigate through the **Project** tab to open the **Compute** menu, and select **Instances**. On the **Instances** dialog, click the **Launch Instance** button to open the **Launch Instance** dialog (shown in Figure 3.5).

**Launch Instance**

Project & User \*   Details \*   Access & Security   Networking \*   Post-Creation

Advanced Options

**Availability Zone**  
nova

**Instance Name \***  
[Empty field]

**Flavor \* ⓘ**  
m1.tiny

**Instance Count \* ⓘ**  
1

**Instance Boot Source \* ⓘ**  
Select source

Specify the details for launching an instance.  
The chart below shows the resources used by this project in relation to the project's quotas.

**Flavor Details**

Name	m1.tiny
VCPUs	1
Root Disk	1 GB
Ephemeral Disk	0 GB
Total Disk	1 GB
RAM	512 MB

**Project Limits**

Number of Instances: 1 of 10 Used

Number of VCPUs: 1 of 20 Used

Total RAM: 2,048 of 51,200 MB Used

Cancel   Launch

Figure 3.5 – The Launch Instance dialog.

Use fields on the **Launch Instance** dialog to describe the EDB Ark instance; on the **Project & User** tab:

- Use the **Project** drop-down listbox to select a tenant for the instance.
- Use the **User** drop-down listbox to the name of the user that will own the instance.

On the `Details` tab:

- Use the `Availability Zone` drop-down listbox to specify an availability zone.
- Use the `Instance Name` field to provide a name for the instance.
- Use the `Flavor` drop-down listbox to specify the size of the cluster. Please note that the cluster must be size `m1.medium` or larger.
- Set the `Instance Count` field to 1.
- Use the drop-down listbox in the `Instance Boot Source` field to select `Boot from image`.
- Use the drop-down listbox in the `Image Name` field to select the name of the image that you imported in Step One.

On the `Access & Security` tab:

- Use the `Key Pair` drop-down listbox to select the keypair you will use to access the instance.
- Check the box next to the name of the security group you created in step 2

On the `Networking` tab

- Select a network from the list of available networks.

No changes are required on the `Post-Creation` and `Advanced Options` tabs.

Click the `Launch` button to launch the console instance.

### 3.4 Step Four – Assign a Floating IP Address

When the instance launch completes, the new instance will be displayed on the Instances panel (as shown in Figure 3.6).

Instance Name	Image Name	IP Address	Size	Key Pair	Status	Availability Zone	Task	Power State	Time since created	Actions
PPCD Console	CentOS 6.6 x86_64	192.168.1.45	m1.small	-	Active	nova	None	Running	0 minutes	Create Snapshot

Figure 3.6– The Instances dialog.

To assign a floating IP address to the new instance, select Associate Floating IP from the drop-down listbox in the Actions column. When the Manage Floating IP Associations dialog opens (see Figure 3.7), use the IP Address drop-down listbox to select an IP address, or click the + button to allocate a new IP address.

Figure 3.7– The Manage Floating IP Associations dialog.

### 3.5 Step Five – Configuring the Installation

Once the console instance has fully launched, you should review the instance log. To review the instance log, log into the OpenStack dashboard as an administrator and navigate into the tenant that is hosting the Ark console. Click the `Project` tab, and then select the `Instances` menu option. Use the drop-down listbox in the `Actions` column) to select `View Log`.

After displaying startup information and SSH authentication details, the log will confirm that the Postgres service and the GlassFish application server have started:

```
Starting postgresql-9.4 service: [ OK ]
Waiting for domain1 to start .....
Successfully started the domain : domain1
domain
Location: /opt/glassfish3/glassfish/domains/domain1
Log File:
/opt/glassfish3/glassfish/domains/domain1/logs/server.log
Admin Port: 4848
Command start-domain executed successfully.
```

After confirming that the service is running, you should use your SSH keypair to SSH to the IP address assigned in Step Four as the user `centos`:

```
ssh -i /path_to_your_private_key centos@ip_address
```

Where:

`path_to_your_private_key` specifies the complete path to the key on your local system. This must be the same key used when launching the console instance (see Section 3).

`ip_address` specifies the floating IP address of the Ark console assigned in Section 3.4.

After connecting, assume the identity of the EDB Ark administrative user:

```
sudo su - cloud_admin
```

Then, use your choice of editor to modify the `ppcd.properties` file. For more information about the `ppcd.properties` file, please see Section 3.5.1.

### 3.5.1 The `ppcd.properties` File

You must supply configuration information before deploying the Ark console on the console host. This information is specified in the `ppcd.properties` file, located in the `/var/ppcd/` directory. Modify the `ppcd.properties` file, specifying the system-specific information detailed below.

Please note that parameter names that start with the word `openstack` have a corresponding value that was declared during the OpenStack installation. The value specified during the OpenStack configuration must match the value specified in the `ppcd.properties` file for EDB Ark to function properly.

#### *PPCD Console DB Backup properties*

Use the parameters in the `PPCD Console DB Backup properties` section to specify backup instructions for the Ark console. By default, the backup properties are commented out; if you uncomment them, the backup service will start when the console application is deployed.

EDB Ark provides a backup script. For console backups to function properly, the console (GlassFish) must be running as the `ppcd` user. The `.pgpass` file for the backup script is located in the `ppcd` user's home directory (`/var/ppcd`).

```
# To enable Console DB Backups, uncomment these properties.
# You must specify console.db.backup.dir and modify the others
# as needed.

# DB user name
# console.db.user=postgres
# DB user password
# console.db.password=c3accd4604514ec58927a462379020fa1f643a0b
# DB name to connect to
# console.db.name=postgres
```

By default, the `console.db.backup.script` parameter specifies the name and location of the script provided with EDB Ark. If you choose to provide your own backup script, use the parameter to specify the name and location. Please note that you must ensure that the script can be read and executed by the `ppcd` OpenStack user.

```
# name of backup script (set to the default script
# shipped with EDB Ark)
# console.db.backup.script=/var/ppcd/.edb/backup-postgresql.sh
```

Use the `console.db.backup.dir` parameter to specify the directory to which console backups will be written. The OpenStack user must have sufficient privileges to write to the specified directory. For information about recovering from a console failure, please see Section 7.

```
# directory to store the backups
# this must be a location that is writeable by the ppcd OS user
# console.db.backup.dir=backup_dir
```

### ***Email Configuration properties***

Use the `contact.email.address` parameter to specify the email address included in the body of cluster status notification emails.

```
# The contact email address that is displayed to the user. This
# is used in cases where the user may need to contact someone
# for more information, e.g. if a user's account is disabled.
```

```
contact.email.address=email_address
```

Use the `email.from.address` parameter to specify the return email address specified on cluster status notification emails.

```
# Return address for all generated emails. This can be
# separate from the mailto links that are included in
# the email bodies.
```

```
email.from.address=email_address
```

Use the `notification.email` parameter to specify the email address to which email notifications about the status of the Ark console will be sent.

```
# the email address that will receive administrative emails from
# the EDB Ark console
notification.email=email_address
```

### ***General properties***

The `wal.archive.container` parameter specifies the name of the object storage container where WAL archives (used for point-in-time recovery) are stored. You must provide a value for this property. Once this property is set, this property must not be changed.

```
# the name of the Object Storage (swift) container used by
# Point-In-Time Recovery (this should never change after
# the initial deployment of EDB Ark).
```

```
wal.archive.container=container_name
```

The `api.timeout` parameter specifies the number of minutes that an authorization token will be valid for use with the API.

```
# the lifetime in minutes of an authorization token used in the
API
api.timeout=10
```

### ***OpenStack specific properties***

Use the `openstack.region` parameter to specify the region in which the Ark console resides. This parameter must match the region specified during OpenStack installation.

```
# the OpenStack region hosting your PPCD console
openstack.region=region_name
```

Use the `openstack.admin.role` parameter to specify the name of the OpenStack administrative role. The OpenStack role is created during the OpenStack installation; when a user that is a member of this role connects to the Ark console, the console will display the Admin and DBA tabs.

```
# the name of the OpenStack admin role
openstack.admin.role=admin_name
```

Use the `openstack.identity.service.endpoint` parameter to specify the URL of the OpenStack Keystone Identity Service.

```
# the URL for the API endpoint for the Identity Service
openstack.identity.service.endpoint=http://identity_service_url
```

Use the `service.account.id` parameter to specify the name of the OpenStack user account that EDB Ark will use when managing clusters. The account must be a member of and be assigned the `admin` role (as specified in the `openstack.admin.role` property) for all tenants that are allowed to run EDB Ark clusters.

Use the `service.account.password` parameter to specify the password associated with the service account.

```
# the account name and password for the EDB Ark service user
# (used internally by EDB Ark)
service.account.id=edbArk_service_user
service.account.password=password
```

### 3.6 Step Six - Deploying the Console

After modifying the `ppcd.properties` file, assume `root` privileges, and use the following command to deploy the Ark console:

```
# /var/ppcd/postInstall.sh
Have you modified the ppcd.properties file according to your
requirements?
Are you sure you want to continue? <y/N> y
Deploying EDB Ark Application...
Application deployed with name PPCDConsole.
Command deploy executed successfully.
Done!
```

The `postInstall.sh` script will remind you to edit `/var/ppcd/ppcd.properties` and prompt you to continue before allowing you to complete installation.

Then, you are ready to log into the Ark console using the IP address assigned to the console in Section 3.4. When connecting to the console, use your OpenStack user name and password and an SSL encrypted browser connection (`https://`). When connecting as an OpenStack Administrative user, EDB Ark Administrative options will be displayed on the Ark console.

In preparation for non-administrative user to connect, an OpenStack Administrator should:

1. Use the Ark console to define a server image for each OpenStack server image that will host a database cluster. For detailed information about using the Ark console to create server images, see Section [4.1.1](#).
2. Use the Ark console to create database engine definitions. For detailed information about defining a database engine, see Section [4.1.2](#).

After deploying the console, OpenStack users that are configured with access permissions may log in; for information about granting access to an OpenStack user, see Section [3.7](#).

### 3.7 Step Seven – Configuring a User to Log In

After deploying the Ark console, the console will be available for connections from enabled OpenStack user accounts. Use the OpenStack console to granted access to an OpenStack user account. Please note that the EDB Ark service account must have administrative privileges in the tenant or project in which you are granting access.

To enable an existing OpenStack user account, connect to the OpenStack console as an Administrative user and select Identity. Click the Manage Members button in the Actions column to the right of the project name.

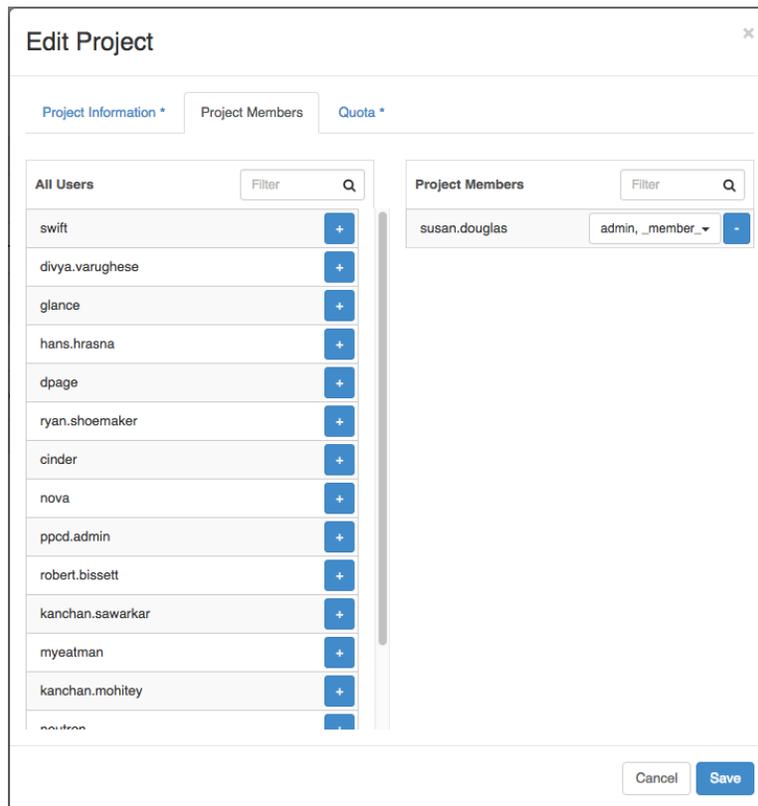


Figure 3.8 – The Edit Project dialog.

The Edit Project dialog opens, displaying the Project Members tab (see Figure 3.8).

- To allow a user to access the project, click the + button to the right of a user's name in the left column. The user will be moved to the right column
- To remove a user's access to a project, click the - button to the right of a user's name in the right column. The user will be moved to the left column.

When you're finished adding users to a project, click `Save` to save your changes and exit the dialog.

### *Creating an OpenStack User with EDB Ark Console Access*

To create an OpenStack user account with access to the Ark console for a specific project, connect to the OpenStack console as a user with Administrative privileges and select `Identity`. Open the `Users` tab, and click the `Create User` button to open the `Create User` dialog (see Figure 3.9).

**Create User** ✕

**User Name \***

**Email**

**Password \***

**Confirm Password \***

**Primary Project \***

 ⌵ +

*Figure 3.9 – The Create User dialog.*

Complete the `Create User` dialog, providing information for the new user:

- Specify the name of the user in the `User Name` field.
- Specify the email address of the user in the `Email` field.
- Specify the password associated with the user account in the `Password` field.

- Re-enter the password in the `Confirm Password` field.
- Use the drop-down listbox in the `Primary Project` field to select the project that will be displayed when the user connects. Please note that the Ark service account must have administrative privileges in the selected project.
- Use the drop-down listbox in the `Role` field to specify if the new role is a `_member_` or an admin user. Please note that `_member_` roles will have sufficient privileges to access the Ark console.

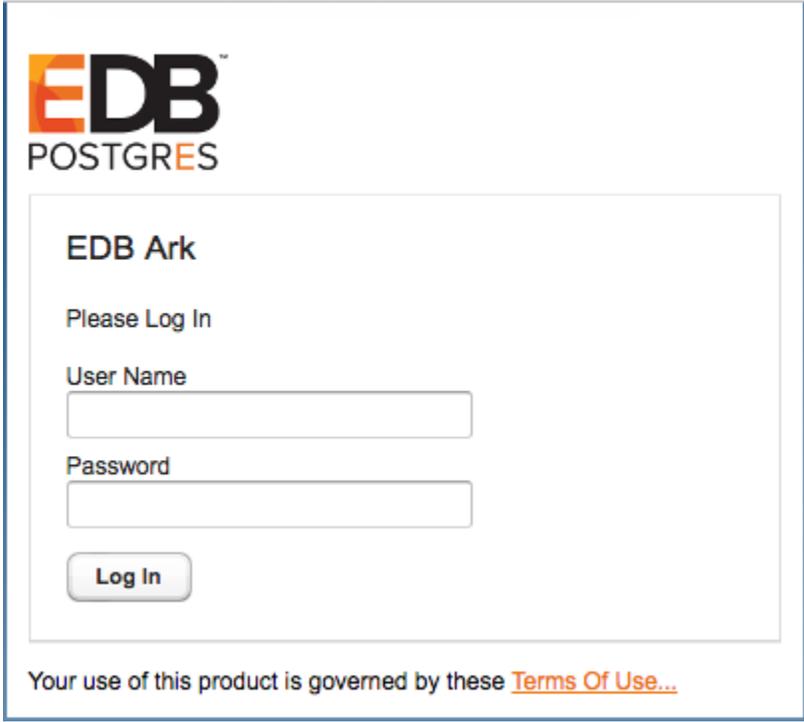
When you've completed the dialog, click the `Create User` button to create the user and exit the dialog. The new user should now be able to access the Ark console

## 4 Administrative Features of the EDB Ark Console

Administrative users have access through the Ark console to features that allow them to register server images and create database engine definitions that will be available for use by the non-administrative EDB Ark user. An administrator also has access to statistical information and console log files that are not available to the non-administrative user.

For information about functionality that is exposed to both administrators and non-administrative users, please see the EDB Ark Getting Started Guide.

When you navigate to the URL of the installed Ark console, the console will display a login dialog (see Figure 4.1).



EDB™  
POSTGRES

EDB Ark

Please Log In

User Name

Password

Log In

Your use of this product is governed by these [Terms Of Use...](#)

*Figure 4.1 - The Login dialog.*

Enter the name of an OpenStack user in the `User Name` field, and the associated password in the `Password` field, and click `Login` to connect to the Ark console. If the user name and password provided are members of the OpenStack administrative role, the Ark console will include the `DBA` tab and the `Admin` tab (as shown in Figure 4.2), and will include a download link for the console logs on the `User` tab.

The screenshot shows the EDB Ark Administrator's console. At the top, there is a navigation bar with tabs for Dashboard, Clusters, Backups, User, DBA, and Admin. The 'User' tab is selected. The main content area is divided into several sections:

- Getting Started:** A box with the text "To begin using EDB Ark, you will want to launch a database cluster." and a button labeled "Launch DB Cluster".
- Resources:** A box with the text "You are using the following resources:" and links for "3 Instances", "2 Snapshots", and "47 Events".
- Hot Topics:** A box with text about PostgreSQL v9.5 and EDB Postgres Advanced Server v9.5, with links to "PostgreSQL 9.5 Release Notes here" and "Advanced Server 9.5 Release Notes here".
- EDB Ark Tutorials and Documentation:** A large heading followed by a grid of links to various PDF guides:
 

Connecting to an EDB Ark Cluster (PDF)	Moving an Existing Database to an EDB Ark Cluster (PDF)	Installing PostGIS in an EDB Ark Cluster (PDF)	Moving an Oracle Database to an EDB Ark Cluster (PDF)
EDB Ark Release Notes	EDB Ark Getting Started Guide (PDF)	EDB Ark Administrative User Guide (PDF)	Installing PEM on an EDB Ark Cluster (PDF)
Advanced Server Guide	PostgreSQL Documentation	Database Compatibility for Oracle(R) Guide	

On the left side, there is a "Contact Us" button, and on the right side, there is a "Free Trials" button.

Figure 4.2 - The EDB Ark Administrator's console.

After connecting to the Ark console, you should:

- Complete the `User` tab, providing a Notification Email. The `User` tab is displayed to any user (administrative or non-administrative). For more information about the `User` tab, see the *EDB Ark Getting Started Guide*.
- Use the `Admin` tab to create the server images and database engines that will be used by non-administrative users. For more information about using the `Admin` tab, see Section [4.1](#).

## 4.1 Using the Admin Tab

Use options on the `Admin` tab (see Figure 4.3) to manage server images and database engines and perform administrative tasks.

The screenshot displays the EDB Ark Admin interface. At the top, there is a navigation bar with tabs for Dashboard, Clusters, Backups, User, DBA, and Admin. The Admin tab is selected. The interface is divided into several sections:

### Server Type Administration

This table allow you to manage base server images which will be provisioned during cluster creation

SERVER ID	SERVER DESCRIPTION	IMAGE ID	INITIAL USER
centos-6.6_x86_64	CentOS 6.6	a8ed57dd-9a34-40ca-977b-ce3af9ad3745	centos
centos-7.1_x86_64	CentOS 7.1	085e925e-557a-424a-acdf-f378370435c6	centos

Buttons: Add Server, Edit Server Details, Delete Server

### DB Engine Administration

This table allows you to manage database engines available for provisioning.

ID	ENABLED	DB TYPE	VERSION	NAME	SERVER TYPE	REQUIRED DB PACKAGES	OPTIONAL NODE PACKAGES
PG_94_C6	true	postgres	9.4	PostgreSQL 9.4 64bit on CentOS/RHEL 6	centos-6.6_x86_64	postgres94-server-pgpool-	
PG_94_C7	true	postgres	9.4	PostgreSQL 9.4 64bit on CentOS/RHEL 7	centos-7.1_x86_64	postgres94-server-pgpool-	
PG_95_C7	true	postgres	9.5	PostgreSQL 9.5 64bit on CentOS/RHEL 7	centos-7.1_x86_64	postgres95-server-pgpool-	
PPAS_94_C6	true	ppas	9.4	Postgres Plus Advanced Server 9.4 64bit on CentOS/RHEL 6	centos-6.6_x86_64	ppas94-server-ppas-pgpool-	
PPAS_94_C7	true	ppas	9.4	Postgres Plus Advanced Server 9.4 64bit on CentOS/RHEL 7	centos-7.1_x86_64	ppas94-server-ppas-pgpool-	
PG_95_C6	true	postgres	9.5	PostgreSQL 9.5 64bit on CentOS/RHEL 6	centos-6.6_x86_64	postgres95-server-pgpool-	pem-agent
PPAS_95_C6	true	ppas	9.5	Postgres Plus Advanced Server 9.5 64bit on CentOS/RHEL 6	centos-6.6_x86_64	ppas95-server-ppas-pgpool-	ppas95-postgis-ppas
PPAS_95_C7	true	ppas	9.5	Postgres Plus Advanced Server 9.5 64bit on CentOS/RHEL 7	centos-7.1_x86_64	ppas95-server-ppas-pgpool-	ppas95-postgis-ppas

Buttons: Add Engine, Edit Engine Details, Delete Engine

### User Administration

Show logged in users

**Wall Message**  
 Display a banner message to all active users and any future users until the message is disabled. The message will persist across console restarts. You can use HTML markup to format the message (<p>, <center>, <a>, etc)

Message:

Buttons: Display Message, Remove Message

### Download Console Logs

Download a zip file of the EDB Ark console log files.

Buttons: Download

Figure 4.3 – The Admin tab

### ***Server and Engine Management***

A fresh installation of EDB Ark will include default DB Engine configurations of:

- EDB Postgres Advanced Server 9.4 and 9.5 (64-bit)
- PostgreSQL 9.4 and 9.5 (64-bit)

The databases (available through the `DB Engine Administration` table) will be disabled and will not have an associated server type or valid repository information. To make a database available for end users, you must:

- Create one or more EDB Ark server images that correspond to an OpenStack server image that resides on your system. For more information about defining a server type, see Section [4.1.1](#).
- Use the `Edit Engine Details` button to modify the existing engine definitions to specify server image associated with the engine and repository information (if applicable), and enable the engine for use by end-users. For more information, see Section [4.1.2](#).

### ***User Administration Features***

Options on the `Admin` tab allow you to:

- Access a list of currently connected users.
- Display a banner message to connected users.
- Download a zip file that contains the console log files.

For more information about `User Administration` features, see Section [4.1.3](#).

### 4.1.1 Managing Server Images

A server definition describes the virtual machine that will host an instance of Advanced Server or PostgreSQL. The definition specifies the identity of an OpenStack server image and the name of the `default_user` specified in the `/etc/cloud/cloud.cfg` file for the image. The image must be either a public image, or available to all tenants that are allowed to run EDB Ark clusters.

#### *Creating a Server Image*

To create a new server image, connect to the Ark console as a user with administrative privileges, navigate to the `Admin` tab, and select `Add Server`. The dialog shown in Figure 4.4 opens.

The screenshot shows a dialog box titled "Add Server" with a close button (X) in the top right corner. Below the title bar is a section header "Server Type Details". There are four input fields: "Server ID", "Server Description", "Image ID", and "Initial User". At the bottom are two buttons: "Save" and "Cancel".

*Figure 4.4 – The Add Server dialog.*

Use the fields on the `Add Server` dialog to define a new server:

- Use the `Server ID` field to provide an identifier for the server image. The `Server ID` must be unique, and may not be modified after saving the server image.
- Use the `Server Description` field to provide a description of the server image.
- Use the `Image ID` field to provide the OpenStack Image ID of the server image.

To locate a server's `Image ID`, connect to the OpenStack administrative console and navigate to the list of `Images`. Select an image name to access the `Image Overview` and locate the image `ID`.

- Use the `Initial User` field to provide the name of the `default_user` that is specified in the `/etc/cloud/cloud.cfg` file for the image. This user must have `sudo root` privileges to perform the initial provisioning of software on the node.

When you have completed the dialog, click `Save` to create the server image, or `Cancel` to exit without saving.

### *Modifying a Server*

Use the `Edit Server Details` button to open the `Edit Server Details` dialog (see Figure 4.5) and modify the properties of a server.

*Figure 4.5 – The Edit Server dialog.*

After modifying the server definition, click `Save` to make the changes persistent and exit the dialog, or `Cancel` to exit without saving.

### *Deleting a Server*

To delete a server definition, highlight a server name, and select the `Delete Server` button. If no engines are dependent on the server, a dialog will open, asking you to confirm that you wish to delete the selected server type (see Figure 4.6).



*Figure 4.6 – The Delete Server Type dialog.*

Select the `Delete` key to remove the server, or `Cancel` to exit without removing the server.

**Error: You can not remove this server type because it is referenced by at least one DB Engine (PPAS\_93-Acctg,PG\_93-Sales)**

*Figure 4.7 – You cannot remove a server with dependencies.*

Please note: If the server is currently used by an engine, the Ark console will advise you that the server cannot be removed (see Figure 4.7); before removing the server, you must delete any dependent engines.

## 4.1.2 Managing Database Engines

An engine definition pairs a Postgres server type with the server image on which it will reside. Only an EDB Ark administrative user can define an engine; once defined, all of the engines that reside within a specific tenant will be made available to all users with access to that tenant.

### *Adding an Engine*

Use the Add Engine dialog (see Figure 4.8) to define an engine. To access the Add Engine dialog, connect to the Ark console as a user with administrative privileges, navigate to the Admin tab, and select Add Engine.

*Figure 4.8 – The Add Engine dialog.*

Use the fields on the Add Engine dialog to define a new server image/database pairing:

- Use the ID field to provide an identifier for the engine. Please note that the identifier must be unique, and may not be modified after saving the engine.
- Use the drop-down listbox in the DB Type field to select the type of database used in the pairing.
- Use the drop-down listbox in the Version field to specify the server version.

- Use the `Name` field to provide a name for the pairing. When the engine is enabled, the specified name will be included for use on the `Create Cluster` dialog.
- Use the drop-down listbox in the `Server Type` field to specify the server image on which the database will reside. The drop-down listbox displays those images previously defined on the `Add Server` dialog.
- Use the `Yum repo URL` field to provide the URL of the yum repository that will be used to initially provision database packages and to later update the database packages during cluster upgrade operations.

The repository URL should take the form:

```
http://[user_name[:password]@]repository_url
```

Where:

`user_name` specifies the name of a user with sufficient privileges to access the repository.

`password` specifies the password associated with the repository user.

`repository_url` specifies the URL of the repository.

Advanced Server updates are available from:

```
http://user_name:password@yum.enterprisedb.com/9.x/redhat/rhel-$releasever-$basearch
```

Advanced Server supporting components are available from:

```
http://user_name:password@yum.enterprisedb.com/tools/redhat/rhel-$releasever-$basearch
```

Please contact your EnterpriseDB account manager for connection credentials (the values specified in the `user_name` and `password` placeholders) for the EnterpriseDB repositories.

PostgreSQL updates (and supporting components) are available from:

```
http://yum.postgresql.org/9.x/redhat/rhel-6-x86_64/pgdg-centos9x-9.x-1.noarch.rpm
```

When specifying multiple repositories in the `Yum repo URL` field, specify one repository per line. When you perform an update, any available updates in all of the specified repositories will be applied.

- Use the `Required DB Packages` field to provide a space-delimited list of packages that have been tested by EDB as the required minimum set to build a functional cluster instance.

When defining a database engine, you must specify the required package list for the installation in the `Required DB packages` field on the `Edit Engine Details` dialog.

For an Advanced Server 9.4 database, the package list must include:

```
ppas94-server
ppas-pgpool134
ppas95-pgpool134-extensions
```

For an Advanced Server 9.5 database, the package list must include:

```
ppas95-server
ppas-pgpool134
ppas95-pgpool134-extensions
```

For a PostgreSQL 9.4 database, the package list must include:

```
postgresql94-server
pgpool-II-94
```

For a PostgreSQL 9.5 database, the package list must include:

```
postgresql95-server
pgpool-II-95
```

Please note that the package list is subject to change.

- Use the `Optional Node Packages` field to provide the names of any packages that should be installed (from the specified repository) on every cluster node during provisioning.

Please note: packages added via the `Optional Node Packages` field on the master node of the cluster will also be provisioned on any standby nodes that are subsequently created. If the package requires manual configuration steps, you will be required to repeat those steps on each node of the cluster; package configurations will not be propagated to standby nodes. If you add a node through cluster operations (such as failover, scaling, or restoring a node from backup), any packages on the new node will require manual configuration.

When you have completed the dialog, click `Save` to create the engine definition, or `Cancel` to exit without saving.

For information about using the EnterpriseDB repository, and the Advanced Server packages available, please see the EDB Postgres Advanced Server Installation Guide, available at:

<http://www.enterprisedb.com/products-services-training/products/documentation/enterpriseedition>

### *Modifying an Engine*

To modify an engine, use the `Edit Engine Details` button to open the `Edit Engine Details` dialog (see Figure 4.9).

*Figure 4.9 – The Edit Engine Details dialog.*

Use fields on the `Edit Engine` dialog to specify property changes to an engine. When you're finished, click the `Save` button to make the changes persistent and exit, or `Cancel` to exit without saving.

### *Disabling an Engine*

You can use the `disabled` box to specify that an engine is (or is not) available for use in new clusters without removing the engine definition:

- If the box next to `disabled` is checked, the engine will not be available for use.
- If the box next to `disabled` is unchecked, the engine will be available for use.

Click the `Save` button to make any changes to the `Edit Engine Details` dialog persistent, or select `Cancel` to exit without modifying the engine definition.

### *Deleting an Engine*

To delete an engine, highlight an engine name in the `DB Engine Administration` list, and select the `Delete Engine` button. A dialog will open, asking you to confirm that you wish to delete the selected engine (see Figure 4.10).



*Figure 4.10 – The Delete DB Engine dialog.*

Click the `Delete` button to remove the engine definition, or select `Cancel` to exit without removing the engine definition.



*Figure 4.11 – The Delete DB Engine dialog.*

Please note that you cannot remove an engine that is referenced by one or more clusters and/or backups; if you attempt to remove an engine that is in use, EDB Ark will display the warning shown in Figure 4.11.

### 4.1.3 Console Administration

Options on the `Admin` tab allow an administrative user to access a list of connected users, to display a message to all connected users, or to download console logs (see Figure 4.12).

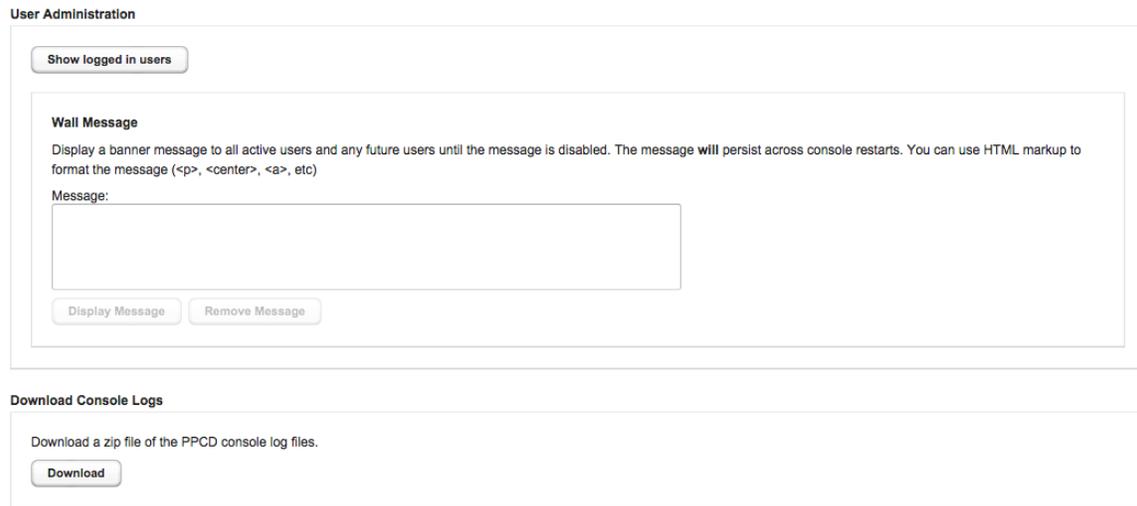


Figure 4.12 –Console administration features on the Admin tab.

#### *Viewing a List of Logged in users*

Select the `Show logged in users` button to display the `Logged in users` dialog (see Figure 4.13).



Figure 4.13 – The Logged in users list.

The dialog displays:

- The current number of empty sessions; an empty session is an http session with the server that is not associated with a logged-in user.
- The current number of sessions with a logged-in user.
- A list of the currently logged-in users.

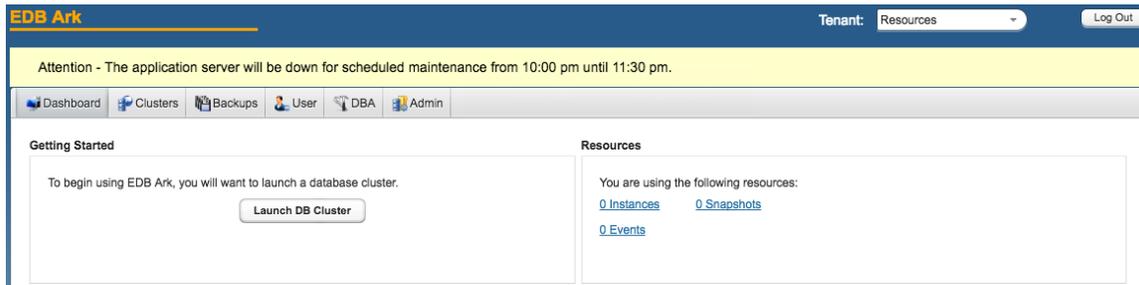
When you're finished reviewing the list, use the x in the upper-right corner of the popup to close the dialog.

### *Modifying the Wall Message*

Use the controls within the Wall Message box (shown in Figure 4.14) to add an announcement to the top of the user console:

*Figure 4.14 - Modifying the Wall Message.*

A message may include HTML tags to control the displayed format, and will wrap if the message exceeds the width of the screen. Enter a message and click the Display Message button to display the banner across the top of each connected user's console (as shown in Figure 4.15)



*Figure 4.15 - Displaying a wall message.*

The message may take a few seconds to refresh. Once processed by the server, the message will be displayed to console users when their screens refresh.

Click the `Remove Message` button to remove the banner. Please note that the wall banner is stored in the console database, and will persist after a server restart; you must use the `Remove Message` button to remove a banner.

### *Accessing the Console Logs*

Use the `Download` button in the `Download Console Logs` panel of the `Admin` tab to download a zip file that contains the server logs for the underlying application server. You can confirm changes to server status or verify server activity by reviewing the application server log file.

EDB Ark stores server log files in:

```
/opt/glassfish3/glassfish/domains/domain1/logs/
```

The most recent server activity is stored in the file named:

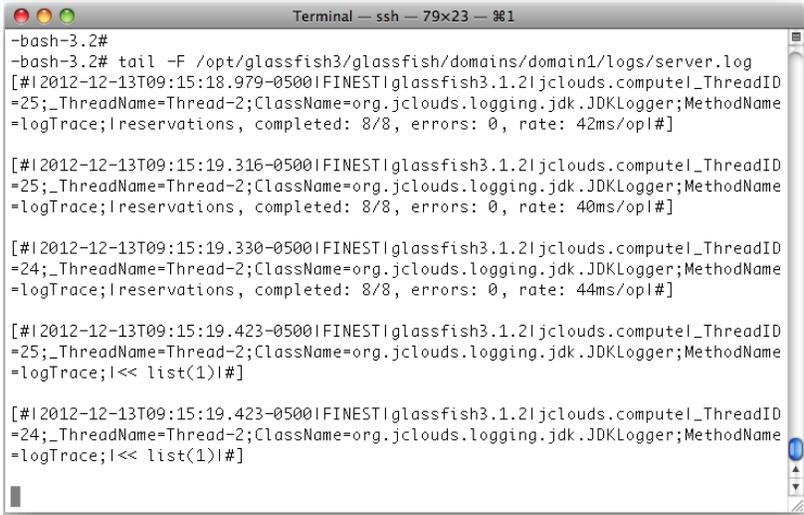
```
server.log
```

When the `server.log` file fills, EDB Ark attaches a unique identifier to the file name, and rotates the file into storage.

You can use the Linux `tail` utility (shown in Figure 4.16) to display the most recent entries in any of the server logs. For example, to review the last 10 lines in the server log file, connect to the console host with `ssh` and enter:

```
tail file_name
```

Where `file_name` specifies the complete path to the log file.



```
Terminal — ssh — 79x23 — 961
-bash-3.2#
-bash-3.2# tail -F /opt/glassfish3/glassfish/domains/domain1/logs/server.log
[#|2012-12-13T09:15:18.979-0500|FINEST|glassfish3.1.2|jclouds.compute|_ThreadID
=25;_ThreadName=Thread-2;ClassName=org.jclouds.logging.jdk.JDKLogger;MethodName
=logTrace;|reservations, completed: 8/8, errors: 0, rate: 42ms/op|#]

[#|2012-12-13T09:15:19.316-0500|FINEST|glassfish3.1.2|jclouds.compute|_ThreadID
=25;_ThreadName=Thread-2;ClassName=org.jclouds.logging.jdk.JDKLogger;MethodName
=logTrace;|reservations, completed: 8/8, errors: 0, rate: 40ms/op|#]

[#|2012-12-13T09:15:19.330-0500|FINEST|glassfish3.1.2|jclouds.compute|_ThreadID
=24;_ThreadName=Thread-2;ClassName=org.jclouds.logging.jdk.JDKLogger;MethodName
=logTrace;|reservations, completed: 8/8, errors: 0, rate: 44ms/op|#]

[#|2012-12-13T09:15:19.423-0500|FINEST|glassfish3.1.2|jclouds.compute|_ThreadID
=25;_ThreadName=Thread-2;ClassName=org.jclouds.logging.jdk.JDKLogger;MethodName
=logTrace;|<< list(1)|#]

[#|2012-12-13T09:15:19.423-0500|FINEST|glassfish3.1.2|jclouds.compute|_ThreadID
=24;_ThreadName=Thread-2;ClassName=org.jclouds.logging.jdk.JDKLogger;MethodName
=logTrace;|<< list(1)|#]
```

*Figure 4.16 - Following the log file with the tail utility.*

You can include the `-F` option to instruct the `tail` utility to display the last 10 lines of the log file, and new log file entries as they are added to the file:

```
tail -F file_name
```

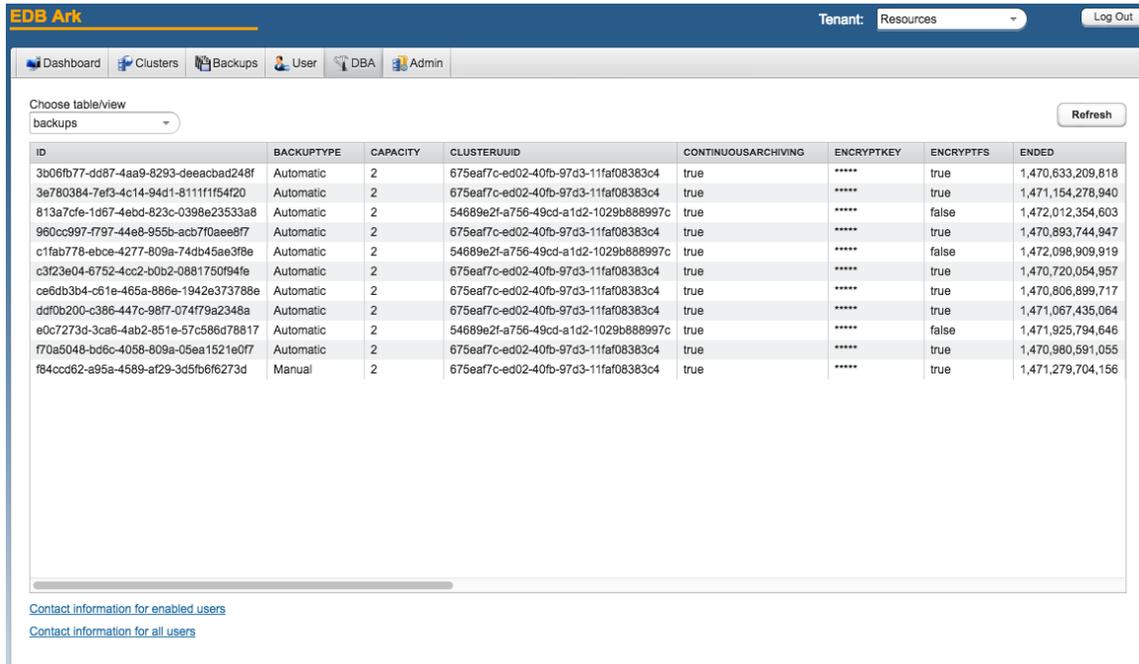
The `tail` utility will continue to display new log file entries if the server log rotates to a new file. Enter `Ctrl-C` to exit `tail` and return to the command prompt.

To review the `tail` command options, enter the command:

```
tail -help
```

## 4.2 Using the DBA Tab

The DBA tab displays views that contain information about current clusters and cluster creation history. The DBA tab (shown in Figure 4.17) is accessible only to administrative users.



The screenshot shows the EDB Ark interface with the DBA tab selected. A dropdown menu is open, showing 'backups' as the selected table/view. Below the dropdown is a table with the following columns: ID, BACKUPTYPE, CAPACITY, CLUSTERUID, CONTINUOUSARCHIVING, ENCRYPTKEY, ENCRYPTFS, and ENDED. The table contains 15 rows of backup data.

ID	BACKUPTYPE	CAPACITY	CLUSTERUID	CONTINUOUSARCHIVING	ENCRYPTKEY	ENCRYPTFS	ENDED
3b06fb77-dd87-4aa9-8293-deeacbad248f	Automatic	2	675eaf7c-ed02-40fb-97d3-11faf08383c4	true	*****	true	1,470,633,209,818
3e780384-7ef3-4c14-94d1-8111f1f54f20	Automatic	2	675eaf7c-ed02-40fb-97d3-11faf08383c4	true	*****	true	1,471,154,278,940
813a7cfe-1d67-4ebd-823c-0398e23533a8	Automatic	2	54689e2f-a756-49cd-a1d2-1029b888997c	true	*****	false	1,472,012,354,603
960cc997-f797-44e8-955b-acb7f0aee8f7	Automatic	2	675eaf7c-ed02-40fb-97d3-11faf08383c4	true	*****	true	1,470,893,744,947
c1fab778-ebce-4277-809a-74db45ae3f8e	Automatic	2	54689e2f-a756-49cd-a1d2-1029b888997c	true	*****	false	1,472,096,909,919
c3f23e04-6752-4cc2-b0b2-0881750f94fe	Automatic	2	675eaf7c-ed02-40fb-97d3-11faf08383c4	true	*****	true	1,470,720,054,957
ce6db3b4-c61e-465a-886e-1942e373788e	Automatic	2	675eaf7c-ed02-40fb-97d3-11faf08383c4	true	*****	true	1,470,806,899,717
ddf0b200-c386-447c-98f7-074f79a2348a	Automatic	2	675eaf7c-ed02-40fb-97d3-11faf08383c4	true	*****	true	1,471,067,435,064
e0c7273d-3ca6-4ab2-851e-57c586d78817	Automatic	2	54689e2f-a756-49cd-a1d2-1029b888997c	true	*****	false	1,471,925,794,646
f70a5048-bd6c-4058-809a-05ea1521e0f7	Automatic	2	675eaf7c-ed02-40fb-97d3-11faf08383c4	true	*****	true	1,470,980,591,055
f84cc062-e95a-4589-af29-3d5fb6f6273d	Manual	2	675eaf7c-ed02-40fb-97d3-11faf08383c4	true	*****	true	1,471,279,704,156

Figure 4.17 - The DBA tab.

Use the Choose table/view drop down listbox (shown in Figure 4.18) to select a view.

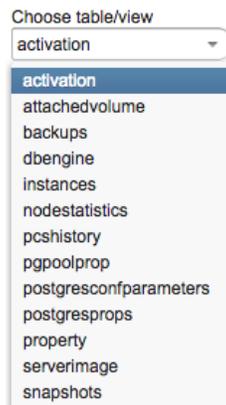


Figure 4.18 - The table/view listbox.

When the view opens, click a column heading to sort the view by the contents of the column; click a second time to reverse the sort order. Use the `Refresh` button to update the contents of the view.

### *Accessing User Information*

Use the user information links in the lower-left corner of the `DBA` tab (shown in Figure 4.19) to download a comma-delimited list of users and user information.

[Contact information for enabled users](#)

[Contact information for all users](#)

*Figure 4.19 - The contact information links*

The file contains the information provided on the `User` tab of the Ark console by each user:

- The user identifier.
- The default email address of the user.
- The first name of the user.
- The last name of the user.
- The company name with which the user is associated.

Select a link to download user information:

- Click `Contact information for enabled users` to download a file that contains only those users that are currently enabled.
- Click `Contact information for all users` to download a file that contains user information of all users (enabled and disabled).

### 4.3 Reference - the DBA Tables

The tables accessed through the DBA tab display a read-only view of the database tables. A DBA can use the information to diagnose some user issues without accessing the console database directly or issuing SQL commands. The tables provide helpful information that a cloud administrator can use when troubleshooting.

For security reasons, the DBA tab does not display the table in which the server stores personal information about registered users, and columns containing sensitive information are obfuscated.

#### 4.3.1 activation

The `activation` table stores the user activation codes that are generated during registration or password recovery. The table contains one entry for each activation code generated.

Column Name	Description
ID	The row identifier for the <code>activation</code> table.
ACTIVATION_TIME	The time that the user activated his account or reset his password.
CODE	A unique code that identifies the transaction. This code is supplied to the user as part of the link in the email.
CODETYPE	The activation code types. The valid types are: NEW_USER RESET_PASSWORD
CREATION_TIME	The time that the activation code was created.
USER_ID	The identity of the user to whom the activation email was sent.

#### 4.3.2 attachedvolume

The `attachedvolume` table provides information about volumes attached to cluster instances. The table contains one entry for each attached volume.

Column Name	Description
ID	The volume to which the instance is attached. The service provider supplies this identifier.
ATTACHTIME	The date and time that the volume was attached.
DEVICE	The mount point of the volume.
INSTANCEID	The cloud service provider's instance identifier.
REGION	The cloud service provider's service region (if applicable).
STATUS	The current status of the volume.

### 4.3.3 backups

The `backups` table provides information about the current backups stored by the server. A backup consists of multiple snapshots (one for each EBS volume in a cluster).

Column Name	Description
ID	A string value that identifies the backup
BACKUPTYPE	Manual Backup if the backup was invoked by a user; Auto Backup if the backup was a scheduled system backup.
CAPACITY	The size of the backup. If the cluster is encrypted, the column will also include <code>(encrypted)</code> .
CLUSTERUUID	The identifier of the cluster from which the backup was created.
CONTINUOUSARCHIVING	True if archiving is enabled; false if archiving is disabled.
ENDED	The time at which the backup ended.
ENGINEVERSION	The Postgres engine version.
MASTERUSER	The name of the database superuser.
NOTES	Notes added by the cluster owner when the snapshot was taken.
OWNER	The name of the cluster owner.
PROGRESS	The most-recent information about the progress of the backup.
SIGNATURE	The name of the cluster owner and the cluster (colon delimited).
STARTED	The time at which the backup began.
WALARCHIVECONTAINER	The name of the archive container in which the WAL logs are stored.
XLOGFILENAME	The identifier of the Xlog file for the backup.
TENANT	The tenant in which the cluster resides.
DBENGINE_ID	The engine number of the database engine used by the cluster.
ENCRYPTFS	True if the content of a backup is stored on an encrypted file system; false if they are not.
YUMUPDATE	True if updates are enabled for the cluster; false if they are not.

### 4.3.4 dbengine

The `dbengine` table provides information about the currently defined database engines. The table contains one entry for each engine.

Column Name	Description
ID	The row ID.
ENGINE_ID	The engine identifier.
EOL	<code>true</code> if the engine is no longer supported; <code>false</code> if the engine is supported.
NAME	The (user-friendly) name of the database engine.
OPTIONAL_PKGS	The optional packages that are installed on the database server (specified in the engine definition).
REQUIRED_PKGS	The required packages that are installed on the database server (specified in the engine definition).
TYPE	The database server type.
VERSION	The version of the database server.
SERVERIMAGE_ID	The database ID of the server image that is linked to the database engine.

### 4.3.5 instances

The `instances` table provides information about the currently active EDB Ark nodes for the EDB Ark service account. The table contains one entry for each instance (master or replica node).

Column Name	Description
ID	The instance ID assigned by the service provider.
AUTOSCALE	<code>true</code> if auto-scaling is enabled on the cluster; <code>false</code> if auto-scaling is disabled.
AVAILABILITYZONE	The data center in which the cluster resides.
BACKUPRETENTION	The number of backups that EDB Ark will retain for the master node of the cluster.
BACKUPWINDOW	The time during which backups will be taken.
CASTATE	The most-recent continuous archiving state of the instance.
CLUSTERNAME	The name of the cluster.
CLUSTERSTATE	The current state of the database. Valid values are: STOPPED = 0 STARTING = 1 RUNNING = 2 WARNING = 3 UNKNOWN = 99
CLUSTERUUID	The unique cluster identifier.
CLUSTERNODEID	On a primary instance, this is the count of how many nodes have been created so far in this cluster, including any dead nodes. On a replica instance, this represents the order in which it was created in the cluster.
CONNECTIONTHRESHOLD	The value specified in the Auto-Scaling Thresholds portion of the Details panel, on the Clusters tab. Specifies the number of connections made before the cluster is scaled up.
CONNECTIONS	The number of active database connections.
CONTINUOUSARCHIVING	Boolean value; <code>true</code> if continuous archiving is enabled, <code>false</code> if continuous archiving is not enabled.
CPULOAD	The current CPU load of the instance.
CPUTHRESHOLD	The CPU load threshold at which the cluster will be automatically scaled up.
CREATIONTIME	The date and time that the node was created.
DATATHRESHOLD	The disk space threshold at which the cluster will be automatically scaled up.
DBNAME	The name of the default database created when the instance was created ( <code>edb</code> or <code>postgres</code> ).
DBPORT	The database listener port.
DBSTATE	The current state of the database: 0 - Stopped 1 - Starting 2 - Running 3 - Warning 99 - Unknown
DNSNAME	The IP address of the instance.
ENGINEVERSION	The version of the database that is running on the instance.
FREEDATASPACE	The current amount of free data space on the instance.
IMAGEID	The server image used when creating the node.

Column Name	Description
INSTANCESTATE	The current state of the node.
IOPS	The requested IOPS setting for the cluster (valid for AWS users only).
MASTERPW	The password of the cluster owner.
MASTERUSER	The name of the cluster owner.
MONITORINGLB	Boolean value; true if load balancing is enabled, false if load balancing is not enabled.
OPTIMIZED	Boolean value; true if an instance is optimized; false if not (valid for AWS users only).
OWNER	The owner of the node.
PARAMETERGROUP	The name of the database parameter group used by the instance.
PENDINGMODIFICATIONS	A message describing any cluster modification in progress (if applicable).
PORT	The SSH port for the cluster.
PRIMARYFAILOVERTOREPLICA	Boolean value; true if the cluster will fail over to a replica; false if the cluster will fail over to a new master instance.
PRIVATEIP	The private IP address of the node.
HARDWARE	The specified hardware size of the instance.
PUBLICIP	The public IP address of the node.
READONLY	True if the node is a read-only replica; false if the node is a master node.
REGION	The region in which the node resides.
SECURITYGROUP	The security group assigned to the node.
SSHKEY	The node's SSH key.
SSHKEYNAME	The name of the node's SSH key.
STORAGE	The amount of disk space on the instance.
SUBNET	The VPC subnet ID (valid for AWS users only).
USEDATASPACE	The current amount of used data space on the instance.
DBENGINE_ID	The selected database engine installed on the instance.
TENANT	The tenant in which the node was created.
VPC	The VPC ID (valid for AWS users only).
YUMSTATUS	The current yum status of the node: 0 - OK 1 - Unknown 2 - Warning 3 - Critical
YUMUPDATE	Boolean value; true if the cluster was created with "yum update" enabled, false if "yum update" was not enabled when the cluster was created.
CLUSTERKEY	The SSH key shared by all of the instances in the cluster.
CLUSTERKEYNAME	The name of the SSH key.
VERSION_NUM	The version of EDB Ark under which the instance was created.
IPPOOL	The name of the floating IP pool (valid for OpenStack users only).
VOLUMETYPE	If supported, the volume type of the cluster.
ENCRYPTFS	True if encryption is enabled for the cluster; false if it is not.
NOTIFICATIONEMAIL	The notification email for the cluster.

### 4.3.6 nodestatistics

The `nodestatistics` table displays information gathered by the cluster manager about the activity for each node. The table contains one record for each time that the cluster manager collected information.

Column Name	Description
ID	The row identifier for the <code>nodestatistics</code> table.
CONNECTIONS	The number of connections to the specified node.
CPULOAD	The processing load placed on the CPU by connecting clients.
FREEMEM	The amount of free memory available to the node.
NODEID	The service provider's node identifier.
OPSPERSECOND	The number of operations per second.
TIMESTAMP	The time at which the data was gathered.
USEDMEM	The amount of used memory (on the node).

### 4.3.7 pcshistory

The `pcshistory` table provides a sortable list of the transactions that have been displayed on the `Events` tabs of the registered users of the EDB Ark service account.

Column Name	Description
ID	The row identifier for the <code>pcshistory</code> table.
CLOCKTIME	The time at which the event occurred.
DESCRIPTION	The description of the event.
OWNER	The registered owner of the cluster on which the event occurred.
SOURCE	The name of the cluster on which the event occurred.

### 4.3.8 pgpoolprop

The `pgpoolprop` table displays the current values assigned to each parameter within each `pgpool.conf` parameter group. Click the `PARAMGROUP` column header to sort the table contents by parameter group.

Column Name	Description
NAME	The name of the parameter within the <code>pgpool.conf</code> file.
OWNER	The owner of the cluster to which the modification has been applied.
ENGINEVERSION	The Postgres engine version.
PARAMGROUP	The name of the parameter group to which the modification was made.
ENABLED	<code>true</code> if the modification has been applied; <code>false</code> if the server is in a default configuration.
PARAMVALUE	The current value of the parameter.

### 4.3.9 postgresconfparameters

The `postgresconfparameters` table provides information about parameter values for the `postgresql.conf` file.

Column Name	Description
NAME	The name of the parameter (within the <code>postgresql.conf</code> file).
ENGINEVERSION	The Postgres engine version.
BOOT_VAL	Reserved for future use.
ENUMVALS	For values of enumerated type, the valid parameter values.
MAX_VAL	The maximum value of the parameter (if limited).
MIN_VAL	The minimum value of the parameter (if limited).
SETTING	The current value of the parameter.
SHORT_DESC	The parameter description.
SOURCE	The source of the parameter value used by the server.
UNIT	A unit of measure (if applicable) for the parameter.
VARTYPE	The data type of the variable.

### 4.3.10 postgresprops

The `postgresprops` table displays the current values assigned to each parameter within each parameter group. Click the `PARAMGROUP` column header to sort the table contents by parameter group.

Column Name	Description
NAME	The name of the parameter (within the <code>postgresql.conf</code> file)
OWNER	The owner of the parameter group.
ENGINEVERSION	The Postgres engine version.
PARAMGROUP	The parameter group in which the parameter is modified.
ENABLED	<code>true</code> if the parameter modification is applied; <code>false</code> if the modification is not applied.
MASTERVERVALUE	The value of the parameter on the master node of the cluster.
REPLICAVALUE	The value of the parameter on the replica node of the cluster.

### 4.3.11 property

The `property` table displays persistent properties used in the console, such as the console name used during backups and wall messages.

Column Name	Description
NAME	The storage location of the console backup.
VALUE	The name of the console.

### 4.3.12 serverimage

The `serverimage` table provides information about currently defined EDB Ark server images.

Column Name	Description
ID	The unique identifier of the server.
IMAGE_ID	The OpenStack identifier of the server image.
INIT_USER	The virtual machine OS user (as provided on the Add Server dialog).
SERVER_DESCRIPTION	The server description.
SERVER_ID	The descriptive identifier of the server.

### 4.3.13 snapshots

The `snapshots` table provides information about cluster backups that reside in the cloud.

Column Name	Description
ID	The unique snapshot identifier.
BACKUPID	An application-managed foreign key reference to the ID column of the <code>backups</code> table.
CAPACITY	The size of the snapshot.
DESCRIPTION	The name of the cluster owner and the cluster (colon delimited).
ENDED	The time at which the backup ended.
ENGINEVERSION	The Postgres engine version.
MASTERPW	The password of the database superuser.
MASTERUSER	The name of the database superuser.
NOTES	Notes added by the cluster owner when the snapshot was taken.
OWNER	The name of the cluster owner.
PROGRESS	The most-recent information about the progress of the snapshot.
SHARED	Deprecated column.
STARTED	The time at which the backup began.
STATUS	Manual Backup if the backup was invoked by a user; Auto Backup if the backup was a scheduled system backup.
VOLUMESIZE	The size of the retained backup.

## 5 Securing EDB Ark

Each cluster has an associated OpenStack security group that specifies the addresses from which the cluster will accept connections. By default, the security group exposes only port 9999 (the load balancing port) to the outside world, while allowing inter-cluster communication, and console-to-cluster communication between the servers in the cluster.

You can modify the security group, strategically exposing other ports for client connection. For example, you may wish to open port 22 to allow `ssh` connections to a server, or port 5444 to allow connections to the listener port of the Advanced Server database server that resides on a replica node.

EDB Ark assigns the same security group to every member of a cluster. By default, the security group contains rules that specify that any cluster member may connect to any other member's ICMP port, TCP port or UDP port. These rules do not permit connections from hosts on the public Internet. You *must not* alter these security rules.

Additional rules open TCP ports 22, 5444 (or 5432) and 7800-7802 to the cluster manager, allowing the cluster manager to perform maintenance and administrative tasks. Please note that the rules governing connections from the cluster manager *must* remain open to allow:

- intra-cluster communications
- communication with the console or cluster manager
- maintenance and administrative functionality

The rule for TCP port 9999 uses a CIDR mask (0.0.0.0/0) to specify that port 9999 is open for connections from any IP address. You can customize this rule, selectively restricting the IP addresses from which computers are allowed to connect to a given port within the cluster.

Please note that EDB Ark provides a secure environment for all communications within the cluster, and between the cluster and the the console or cluster manager by employing SSH authentication and encryption.

## 5.1 Opening a Port for SSH Connections

Before a user may SSH to a node on an EDB Ark cluster, an OpenStack Administrative user must modify the cluster's security group to allow the connection.

To access a list of security groups for the currently running clusters, connect to the OpenStack console and select **Access & Security** from the **Compute** menu. Click the **Manage Rules** button to the right of a cluster name to view detailed security group rules for the cluster (see Figure 5.1).

<input type="checkbox"/>	Direction	Ether Type	IP Protocol	Port Range	Remote IP Prefix	Remote Security Group	Actions
<input type="checkbox"/>	Egress	IPv6	Any	Any	:::0	-	Delete Rule
<input type="checkbox"/>	Egress	IPv4	Any	Any	0.0.0.0/0	-	Delete Rule
<input type="checkbox"/>	Ingress	IPv4	TCP	22 (SSH)	192.168.1.20/32	-	Delete Rule
<input type="checkbox"/>	Ingress	IPv4	TCP	22 (SSH)	192.168.1.19/32	-	Delete Rule
<input type="checkbox"/>	Ingress	IPv4	TCP	22 (SSH)	172.16.251.3/32	-	Delete Rule
<input type="checkbox"/>	Ingress	IPv4	TCP	5432	192.168.1.20/32	-	Delete Rule
<input type="checkbox"/>	Ingress	IPv4	TCP	5432	192.168.1.19/32	-	Delete Rule
<input type="checkbox"/>	Ingress	IPv4	TCP	7800 - 7802	192.168.1.19/32	-	Delete Rule
<input type="checkbox"/>	Ingress	IPv4	TCP	7800 - 7802	172.16.251.8/32	-	Delete Rule
<input type="checkbox"/>	Ingress	IPv4	TCP	7800 - 7802	172.16.251.3/32	-	Delete Rule
<input type="checkbox"/>	Ingress	IPv4	TCP	7800 - 7802	192.168.1.20/32	-	Delete Rule
<input type="checkbox"/>	Ingress	IPv4	TCP	9999	0.0.0.0/0	-	Delete Rule

Displaying 12 items

*Figure 5.1 – Detailed security rules for a cluster.*

To add a rule that opens a port for ssh connections to a cluster, click the **Add Rule** button in the upper-right corner of the **Manage Security Groups** window. When the **Add Rule** dialog opens, use the drop-down listbox in the **Rule** field to select **SSH**.

**Add Rule**

Rule \*  
SSH

Remote \* ⓘ  
CIDR

CIDR ⓘ  
0.0.0.0/0

**Description:**  
Rules define which traffic is allowed to instances assigned to the security group. A security group rule consists of three main parts:

**Rule:** You can specify the desired rule template or use custom rules, the options are Custom TCP Rule, Custom UDP Rule, or Custom ICMP Rule.

**Open Port/Port Range:** For TCP and UDP rules you may choose to open either a single port or a range of ports. Selecting the "Port Range" option will provide you with space to provide both the starting and ending ports for the range. For ICMP rules you instead specify an ICMP type and code in the spaces provided.

**Remote:** You must specify the source of the traffic to be allowed via this rule. You may do so either in the form of an IP address block (CIDR) or via a source group (Security Group). Selecting a security group as the source will allow any other instance in that security group access to any other instance via this rule.

Add

Figure 5.2 – Opening a port for an SSH connection.

When you select SSH, the Add Rule dialog will change to display only those fields that are required to define a rule that allows an SSH connection (see Figure 5.2). Use the fields to specify your connection preferences:

- Use the Remote drop-down listbox to specify the type of traffic that will be allowed to connect via this rule. The connection options for an SSH rule are CIDR and Security Group; the default is CIDR.
- Specify who may connect via the new rule:

If you selected CIDR, provide the CIDR-formatted address or addresses that are allowed to connect to the server via ssh. By default, the OpenStack console displays the address 0.0.0.0/0, opening port 22 for connections from any host.

For more information about specifying a CIDR address, see:

<http://www.postgresql.org/docs/9.5/static/datatype-net-types.html>

If you selected Security Group, use the Security Group and Ether Type drop-downs to make the appropriate system-specific selections.

## 5.2 Using iptables Rules

If you are using iptables rules to manage security in an OpenStack image or on the host of the Ark console, please note that you must not modify the iptables rules provided by EDB Ark.

If you are using iptables on the host of the Ark console, do not modify the following rules:

```
iptables -A PREROUTING -t nat -i eth0 -p tcp --dport 80 -j
    REDIRECT --to-port 8080
iptables -A PREROUTING -t nat -i eth0 -p tcp --dport 443 -j
    REDIRECT --to-port 8181
iptables -I INPUT 1 -p tcp --dport 8181 -j ACCEPT
iptables -I INPUT 1 -p tcp --dport 8080 -j ACCEPT
/sbin/service iptables save
```

These rules:

- redirect http and https traffic on ports 80 and 443 to the default GlassFish ports (8080 and 8181).
- allow inbound traffic on 8080 and 8181.
- save the configuration (to preserve the behaviors when the system reboots).

If you are using iptables on an Advanced Server cluster, do not modify the following rules:

```
iptables -I INPUT 1 -p tcp --dport 7800:7802 -j ACCEPT
iptables -I INPUT 1 -p tcp --dport 5444 -j ACCEPT
iptables -I INPUT 1 -p tcp --dport 9999 -j ACCEPT
/sbin/service iptables save
```

If you are using iptables on a PostgreSQL cluster, do not modify the following rules:

```
iptables -I INPUT 1 -p tcp --dport 7800:7802 -j ACCEPT
iptables -I INPUT 1 -p tcp --dport 5432 -j ACCEPT
iptables -I INPUT 1 -p tcp --dport 9999 -j ACCEPT
/sbin/service iptables save
```

The rules:

- allow inbound traffic from the Ark console on ports 7800 and 7802.
- allow inbound traffic on the database listener ports.
- save the configuration (to preserve the behaviors when the system reboots).
- allow inbound traffic on the load balancer port.

## 6 Console Management

The sections that follow provide information about managing the EDB Ark application server.

### 6.1 Starting, Stopping or Restarting the Server

The application server behind the Ark console is GlassFish. The service runs as a user named `ppcd`; before invoking any commands that change the state of the service, you must assume the identity of `ppcd`.

To stop, start or restart the application server, use `ssh` to connect to the host of the Ark console database as a user with `sudo` privileges. Then, assume the identity of `ppcd`:

```
sudo su - ppcd
```

Then, to start the server:

```
/opt/glassfish3/glassfish/bin/asadmin start-domain
```

To stop the server:

```
/opt/glassfish3/glassfish/bin/asadmin stop-domain
```

To restart the server (if it is already running):

```
/opt/glassfish3/glassfish/bin/asadmin restart-domain
```

If prompted, provide the password associated with the GlassFish administrator account. For more information about setting the GlassFish administrator password, see Section [6.4](#).

## 6.2 Upgrading the Console

The steps that follow provide detailed instructions about upgrading the Ark console. Before upgrading the console, you must ensure that no users are connected to the console, and that there are no cluster operations (backup, cloning, etc) in progress. You may wish to alert users to the pending upgrade with a wall message; for details about setting a wall message, see Section 4.1.3.

Use the `Show logged in users` button on the `Admin` tab to confirm that no users are connected to the console, and check the server log (located in `/opt/glassfish3/glassfish/domain1/logs/server.log`) to confirm that all server activities have completed.

1. After confirming that the system is not in use, use `ssh` to connect to the node on which the Ark console resides, and assume root privileges:

```
sudo su -
```

2. With your choice of editor, modify the repository configuration file (located in `/etc/yum.repos.d`), adding your connection credentials to the `edb-ark` repository URL:

```
[edb-ark]
name=EnterpriseDB EDB Ark
baseurl=http://user_name:password@yum.enterprisedb.com/edb-ark/redhat/rhel-$releasever-$basearch
enabled=0
gpgcheck=0
gpgkey=file:///etc/pki/rpm-gpg/ENTERPRISEDB-GPG-KEY
```

To enable the repository, replace the `user_name` and `password` placeholders with your user name and password, and set `enabled` to 1.

3. Use the `yum list "edb-ark*"` command to review a list of available updates.

```
yum list "edb-ark*"
```

4. If any updates are available, use `yum` to install the updates:

```
yum update package_name
```

Where `package_name` specifies the name of the package that you wish to update.

5. When the downloads complete, navigate into the `/var/ppcd` directory:

```
cd /var/ppcd
```

6. Invoke the EDB Ark installation script to upgrade the console:

```
./postInstall.sh
```

The installation script will prompt you to confirm that the console is not in use, and that you wish to continue with the installation.

7. Enter `y` to perform the console upgrade.

## 6.3 Customizing the Console

The majority of the console layout is defined in source files and cannot be changed without compilation, but you can modify several aspects of the user interface, including:

- Background images
- Background colors
- Fonts
- Font colors

To change the colors, fonts, or images displayed by the console, you can use `ssh` to connect to the console host; once connected, use your choice of editor to modify the files that control the onscreen display.

### *Modifying the Console Display*

To modify the console display, use `ssh` to connect to the host of the Ark console: After connecting to the console host, you can use your choice of editor to modify the files that control the look and feel of the console host.

***Please Note: We recommend that you make a backup of any file that you plan to modify before changing the file.***

### *The css File*

The `css` rules for the EDB Ark user console are stored in the `styles.css` file. The file is located at:

```
/opt/glassfish3/glassfish/domain1/applications/PPCDConsole/
VAADIN/themes/pcsconsole/styles.css
```

Please refer to comments within the file for detailed information about modifying individual components within the console display.

Some modifications to the `styles.css` file will be visible when you reload the page in your browser; if a change is not immediately visible, restart the server to apply the changes. If a change is not visible after restarting the server, you may need to clear your browser cache.

### *The images Directory*

To modify the images that are displayed by the console user interface, replace the `.png` files in the `images` directory with the images you wish to display. The `images` directory is located at:

```
/opt/glassfish3/glassfish/domains/domain1/applications/PPCD  
Console/VAADIN/themes/pcsconsole/images
```

Please note that the logo displayed on the login screen is defined in the `i18n.properties` file; for more information about modifying the logo image, please refer to comments in that file.

### ***The html Template File***

The `loginscreen.html` template file defines the page layout for the login screen and the terms of use URL (referenced on the login screen). The file is located at:

```
/opt/glassfish3/glassfish/domains/domain1/applications/PPCDConsole/  
WEB-INF/classes/com/enterprisedb/pcs/ui/loginscreen.html
```

### ***The properties File***

Use the `i18n.properties` file to modify text and external URLs displayed in the Ark console. The `i18n.properties` file is located at:

```
/opt/glassfish3/glassfish/domains/domain1/applications/PPCD  
Console/WEB-INF/classes/i18n.properties
```

Comments within the `i18n.properties` files identify the onscreen information controlled by each entry in the file. You must restart the server to apply any modifications to the `properties` file.

## 6.4 Changing Console Passwords

Each fresh installation of the console uses the same default passwords; after installing the console, you should modify the passwords used by the console to create a more secure environment.

### *Modifying the PostgreSQL Database User's Password*

A fresh installation of the Ark console includes a PostgreSQL installation that is used to manage the console; the management database is named `postgres`. By default, the database superuser has the following connection credentials:

```
name: postgres
password: c3accd4604514ec58927a462379020fa1f643a0b
```

You should change the database superuser's password on the PostgreSQL server to a unique password (known only to trusted users). After changing the superuser password on the PostgreSQL database, you will need to copy that password to the JDBC connection pool. If you have enabled console backups, you must also modify the `ppcd.properties` file and the `.pgpass` file.

You can use the SQL `ALTER ROLE` command and the `psql` client to change the password on the Postgres server. To start the `psql` client, the `bin` directory must be in your search path. At a terminal window, connect to the `psql` client with the command:

```
psql -d postgres -U postgres
```

When prompted, supply the password of the `postgres` database user. After connecting to the database, you can use the `ALTER ROLE` command to modify the password associated with the `postgres` user:

```
ALTER ROLE postgres password 'new_password';
```

Where:

*new\_password* is the new password of the `postgres` role.

After modifying the password associated with the database superuser, use the `\q` meta-command to exit the `psql` client.

After changing the password of the database superuser, you must also change the password in the JDBC connection pool. At the command line, assume the identity of the `ppcd` user:

```
sudo su - ppcd
```

Then, use the following `asadmin` utility to modify the password:

```
asadmin set resources.jdbc-connection-pool.pcsconfig-
pool.property.password=new_password
```

Where:

*new\_password* is the password of the postgres role.

After modifying the password for the JDBC connection pool, you can ping the connection pool to test the jdbc connector. Use the command:

```
asadmin ping-connection-pool pcsconfig-pool
```

If the ping is successful, the command will return:

```
Command ping-connection-pool executed successfully.
```

If the ping is not successful, the command will return:

```
remote failure: Ping Connection Pool failed for pcsconfig-
pool.
Connection could not be allocated because: FATAL: password
authentication failed for user "postgres" Please check the
server.log for more details.
```

If you have enabled console backups, you must also modify the `ppcd.properties` files, adding the new password:

```
console.db.password=new_password
```

For information about modifying the `ppcd.properties` file, see Section [3.5.1](#).

Then, modify the `.pgpass` file, replacing the old password associated with the postgres role with the new password. By default, the `.pgpass` file is located in the home directory of the `ppcd` user (`~ppcd`). Use your choice of editor to modify the `.pgpass` file, updating the password.

For more information about modifying the `.pgpass` file, please see:

<http://www.postgresql.org/docs/9.5/static/libpq-pgpass.html>

### *Modifying the GlassFish Console Password*

By default, the GlassFish console user has the following connection credentials:

```
name: admin
password: ChangeIt2015!
```

To modify the password associated with the GlassFish user, use `ssh` to connect to the console image, authenticating yourself with the account id and key pair used when the instance was created (See Section [3.3](#)). Then, assume the identity of the `ppcd` user:

```
sudo su - ppcd
```

Then, use the `asadmin` utility to change the password (see Figure 6.1). The utility will prompt you through the process of resetting your password:

```
asadmin change-admin-password
Enter admin user name [default: admin]>
```

Provide the name of the administrative user and press `Return`.

```
Enter admin password>
```

Provide the password associated with the administrative user and press `Return`; by default, the password is `ChangeIt2015!`.

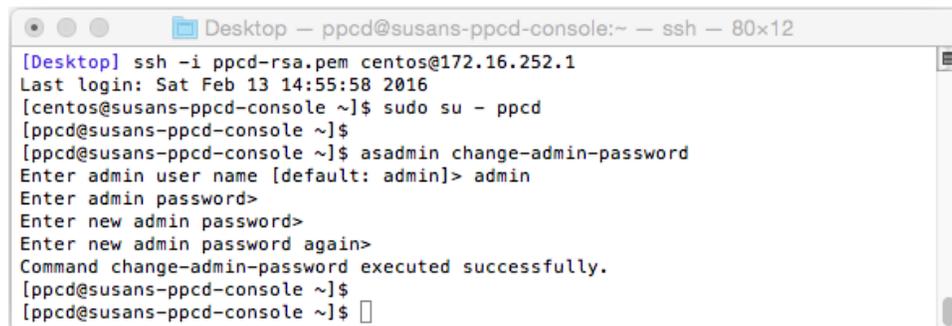
```
Enter new admin password>
```

Enter a new password for the console user and press `Return`.

```
Enter new admin password again>
```

Confirm the new password, and press `Return`. The `asadmin` utility will confirm:

```
Command change-admin-password executed successfully.
```

A screenshot of a terminal window titled "Desktop - ppcd@susans-ppcd-console:~ - ssh - 80x12". The terminal shows the following sequence of commands and prompts:

```
[Desktop] ssh -i ppcd-rsa.pem centos@172.16.252.1
Last login: Sat Feb 13 14:55:58 2016
[centos@susans-ppcd-console ~]$ sudo su - ppcd
[ppcd@susans-ppcd-console ~]$
[ppcd@susans-ppcd-console ~]$ asadmin change-admin-password
Enter admin user name [default: admin]> admin
Enter admin password>
Enter new admin password>
Enter new admin password again>
Command change-admin-password executed successfully.
[ppcd@susans-ppcd-console ~]$
[ppcd@susans-ppcd-console ~]$
```

*Figure 6.1 – Changing the console user's password.*

If you use the `asadmin` utility often (for example, starting and stopping the console server), you can use the `asadmin login` command to save the credentials for the current connected user. Use `ssh` to connect to the console image, and invoke the command:

```
asadmin login
```

The utility will prompt you for authentication information:

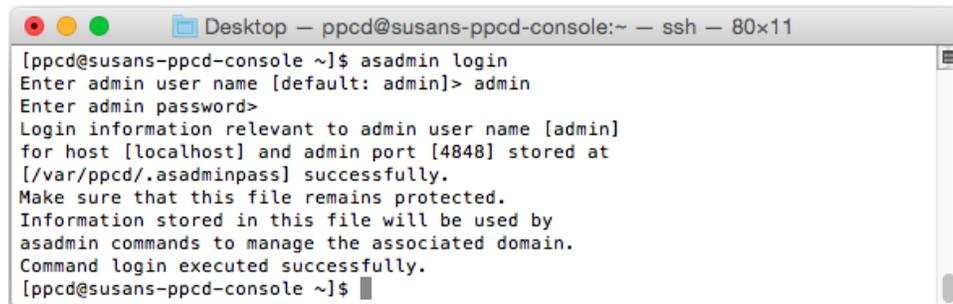
```
Enter admin user name [default: admin]>
```

Provide a user name and press `Return`.

```
Enter admin password>
```

Provide the password associated with the user, and press `Return`. The console will respond:

```
Login information relevant to admin user name [admin]
for host [localhost] and admin port [4848] stored at
[/var/ppcd/.asadminpass] successfully.
Make sure that this file remains protected.
Information stored in this file will be used by
asadmin commands to manage the associated domain.
Command login executed successfully.
```



```
Desktop - ppcd@susans-ppcd-console:~ - ssh - 80x11
[ppcd@susans-ppcd-console ~]$ asadmin login
Enter admin user name [default: admin]> admin
Enter admin password>
Login information relevant to admin user name [admin]
for host [localhost] and admin port [4848] stored at
[/var/ppcd/.asadminpass] successfully.
Make sure that this file remains protected.
Information stored in this file will be used by
asadmin commands to manage the associated domain.
Command login executed successfully.
[ppcd@susans-ppcd-console ~]$
```

*Figure 6.1 – Invoking `asadmin login`.*

## 7 Recovering From a Console Failure

You can use parameters in the `ppcd.properties` file to configure EDB Ark to backup the console database hourly. If you do not wish to implement backups with the Ark backup script, you should maintain regular backups of your console database.

All of the tables used by the Ark console are stored in the `postgres` database. You can use the Postgres `pg_dump` or `pg_dumpall` command to archive the console database and preserve that information. If necessary, you can then use the `pg_restore` command to restore the console database.

If the console application or application server should fail, the user and instance information will persist in the console database, and will be available when the console application restarts. If the virtual machine hosting the application and database fails, then all information about the console database and registered users will be lost. While user and instance information will be lost, user clusters will still be operational, but without support or management from the Ark console.

### 7.1 Enabling Console Backups with the EDB Ark Script

For the console backup script to function properly, the console (GlassFish) must be running as the `ppcd` user, and the `ppcd` user must have sufficient privileges to read and execute the backup script. The `.pgpass` file (used for backup authentication) is located in the `ppcd` user's home directory (`/var/ppcd`).

Use the parameters in the `PPCD Console DB Backup properties` section of the `ppcd.properties` file to specify backup instructions for the Ark console. By default, the backup properties are commented out; when you uncomment the parameters, the backup service will start when the console application is deployed.

```
# To enable Console DB Backups, uncomment these properties.
# You must specify console.db.backup.dir and modify the others
# as needed.

# DB user name
# console.db.user=postgres
# DB user password
# console.db.password=postgres
# DB name to connect to
# console.db.name=postgres
```

By default, the `console.db.backup.script` parameter specifies the name and location of the script provided with EDB Ark. If you choose to provide your own backup

script, use the parameter to specify the name and location. Please note that you must ensure that the script can be read and executed by the `ppcd` user.

```
# name of backup script (set to the default script
# shipped with PPCD)
# console.db.backup.script=/var/ppcd/.edb/backup-postgresql.sh
```

Use the `console.db.backup.dir` parameter to specify the directory to which backups will be written. The `ppcd` user must have sufficient privileges to write to the specified directory.

The backup directory specified should not reside on the console VM's root disk; your backup would be lost in the event of a VM failure. You should consider mounting an external volume to the console VM, and writing console database backups to that volume.

```
# directory to store the backups
# this must be a location that is writeable by the ppcd user
# console.db.backup.dir=backup_dir
```

## 7.2 Recovering the Console

The backup script provided with the Ark console uses `pg_dump` to create a plain-text SQL script file that contains the commands required to rebuild the console database to the state in which the backup was taken. You can use the following command to invoke the `psql` command line tool and use the script to restore the console:

```
/usr/bin/psql -h localhost -p 5432 -d postgres -U postgres
-f <(echo truncate sequence\;; cat recovery_file
```

Where *recovery\_file* specifies the path and name of the backup file you wish to restore.

While restoring a console instance, you should shut down the application server so that the console application isn't actively using the database. When the restoration is complete, restart the application server.

## 8 Notifications

EDB Ark will send e-mail notifications when:

- The state of a monitored database cluster changes.
- An administrative action is performed on a cluster
- User information changes.

Please note: For EDB Ark notifications to function properly, you must have an SMTP server running on each node, and specify the administrator's email address in the `ppcd.properties` file and the cluster owner's email address in the Ark console.

Subject	Body
Console DB Backup Failed	The Console DB Backup failed. A problem was encountered trying to run the backup script: <code>script_output</code> .
Database State Changed to <code>db_state</code>	The MASTER REPLICA database server <code>dns_name</code> in cluster <code>cluster_name</code> is now STOPPED STARTING RUNNING WARNING UNKNOWN in location <code>availability_zone</code> .
Load Balancer Port Error	The MASTER REPLICA database server <code>dns_name</code> in cluster <code>cluster_name</code> in location <code>availability_zone</code> is reporting an error determining the load balancer port.
Load Balancer Port Notification	The MASTER REPLICA database server <code>dns_name</code> in cluster <code>cluster_name</code> is now RUNNING STARTING STOPPED WARNING UNKNOWN in location <code>availability_zone</code> using port <code>port_number</code> .
Continuous Archiving State Changed to <code>db_state</code>	Continuous Archiving on the master replica database server <code>dns_name</code> in cluster <code>cluster_name</code> is operating normally.
Continuous Archiving State Changed to <code>db_state</code>	A problem was detected with continuous archiving on the master replica database server <code>dns_name</code> in cluster <code>cluster_name</code> .
Data Storage Scaling <code>cluster_name</code>	Data storage is being added to cluster <code>cluster_name</code> because the auto-scaling threshold was reached.

Data storage scaling for cluster <i>cluster_name</i> has been suspended	Data storage scaling for cluster <i>cluster_name</i> has been suspended. Instance <i>instance_id</i> no assignable device names left
Rebuild of primary node in cluster <i>cluster_name</i>	The primary server, node id <i>instance_id</i> in cluster <i>cluster_name</i> is being rebuilt.
Replacement of primary node in cluster <i>cluster_name</i>	The primary server, node id <i>instance_id</i> in cluster <i>cluster_name</i> is being replaced with node id <i>instance_id</i> .
Rebuild of replica node in cluster <i>cluster_name</i>	The replica server, node id <i>instance_id</i> in cluster <i>cluster_name</i> is being rebuilt.
Replica promotion failed in cluster <i>cluster_name</i>	Replica promotion failed. Performing rebuild of primary DB node; id: <i>instance_id</i>
Replica promotion failed in cluster <i>cluster_name</i>	Replica promotion failed. Node id: <i>instance_id</i>
WARNING: Connectivity Issue with instance <i>region / instance_id</i>	WARNING: The EDB Ark cluster manager was unable to connect to the node manager for instance ID <i>region/instance_id</i> . This may be due to a temporary connectivity issue or the instance may require manual intervention.
(PITR) Base Backup of cluster <i>cluster_name</i> failed	The automatic manual backup of cluster <i>cluster_name</i> in location <i>availability_zone</i> failed.
Backup of cluster <i>cluster_name</i> failed	The automatic manual backup of cluster <i>cluster_name</i> in location <i>availability_zone</i> failed.
WAL Archive Storage	A storage container (bucket) named <i>bucket_name</i> has been created. All EDB Ark clusters configured for Continuous Archiving (Point-in-Time

Container Created	Recovery) will use this location to store archived WAL files. This container should not be deleted once created as it will cause WAL archiving to stop functioning.
Termination of cluster <i>cluster_name</i> completed.	The termination of cluster <i>cluster_name</i> has completed.
WARNING: Termination Protection <i>instance_id</i> .	The system was not able to terminate instance {0} in cluster <i>cluster_name</i> because termination protection is enabled. You must disable termination protection before this instance can be terminated.
OS/SW update PASSED on node <i>instance_id</i> .	Yum update results for node: <i>dns_name</i> Yum exit status: <i>exit_status</i> You may also consult the yum log on the node (usually in <i>/var/log/yum.log</i> ) If there were any errors, you will have to log into the node and manually correct them and/or consult with your EDB Ark Admin.
OS/SW update FAILED on node <i>instance_id</i> .	Yum update results for node: <i>dns_name</i> Yum exit status: <i>exit_status</i> You may also consult the yum log on the node (usually in <i>/var/log/yum.log</i> ) If there were any errors, you will have to log into the node and manually correct them and/or consult with your EDB Ark Admin
OS/SW Status is now: <i>status</i>	The OS/SW status on node <i>dns_name</i> of cluster <i>cluster_name</i> is now CRITICAL. This indicates that the node has at least one outstanding security update and possibly other non-critical updates available. Please log into the EDB Ark console and perform a cluster upgrade.
OS/SW Status is now: <i>status</i>	The OS/SW status on node <i>dns_name</i> of cluster <i>cluster_name</i> is now UNKNOWN. This indicates that the node is having difficulty determining the OS/SW status. This may be a temporary issue that will resolve itself. Please log into the EDB Ark console and check your cluster's status. If it is still showing status UNKNOWN then you will need to log into node <i>dns_name</i> and run "yum --security check-update" to diagnose the issue manually.
Unable to delete Security Group <i>group_name</i> .	The system was not able to delete the Security Group named <i>group_name</i> in cluster <i>cluster_name</i> . This could be because one or more instances in the cluster could not be terminated. This Security Group will need to be manually

	deleted from the provider's management console.
Volume attachment failed in cluster <i>cluster_name</i>	The message body contains error text directly from OpenStack
Reboot of cluster <i>cluster_name</i> in progress	OS/SW update completed successfully, rebooting all cluster nodes.

## 9 Resources

You can also find solutions to administrative problems through EnterpriseDB:

If you have purchased support, you can log a support ticket:

- in the Customer Portal: <http://www.enterprisedb.com/support>
- via email: <mailto:support@enterprisedb.com>
- or by phone: +1-732-331-1320 or 1-800-235-5891 (US Only)

If you have not purchased support, and would like to, view your support options at:

<http://www.enterprisedb.com/cloud-database/support>

You are always welcome to log an issue via email; when time permits, our customer support experts will respond to inquiries from customers that have not purchased support.

You can also find free help on a wide variety of topics in the EnterpriseDB User Forums, at:

<http://forums.enterprisedb.com/forums/show/21.page>

Postgres documentation and helpful tutorials are available from the EDB Ark bookshelf, located on the `Dashboard` tab of the management console.

### 9.1 Licenses

License files for EDB Ark and supporting third-party libraries are located in the root filesystem:

```
/EDBArk_3rd_party_licenses.txt
```

```
/EDBArk_license.txt
```