EDB™ Ark

Administrative User's Guide

Version 2.1

February 9, 2017

EDB™ Ark Administrative User's Guide, Version 2.1
by EnterpriseDB Corporation
Copyright © 2017 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 **www**.enterprisedb.com

Table of Contents

Copyright © 2011 - 2017 EnterpriseDB Corporation. All rights reserved.

Copyright © 2011 - 2017 EnterpriseDB Corporation. All rights reserved.

Copyright © 2011 - 2017 EnterpriseDB Corporation. All rights reserved.

1 Introduction

EDB Ark automatically provisions EDB Postgres Advanced Server or PostgreSQL databases in single instances, high-availability clusters, or application development sandboxes in an Amazon Web Services (AWS) AMI or in an OpenStack 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.

Copyright © 2011 - 2017 EnterpriseDB Corporation. All rights reserved.

- **Console Management** - Section 5.5 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.

- **AWS Policies** – Section 10 provides security and trust policies required when creating AWS user accounts.

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.

Copyright © 2011 - 2017 EnterpriseDB Corporation. All rights reserved.

1.1 What's New

The following features have been added to EDB Ark for release 2.1:

- The Ark console can now create and manage PostgreSQL and Advanced Server clusters in an Amazon public cloud or on an OpenStack host.

- The Ark console now features the `Consoles` drop-down listbox. The `Consoles` drop-down provides an administrator-customized shortcut that opens another browser tab and navigates to other Cloud consoles.

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`".

Copyright © 2011 - 2017 EnterpriseDB Corporation. All rights reserved.

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 pgpool-II to implement load balancing and connection pooling among all active instances. Figure 2.1 provides a general overview of the EDB Ark architecture.

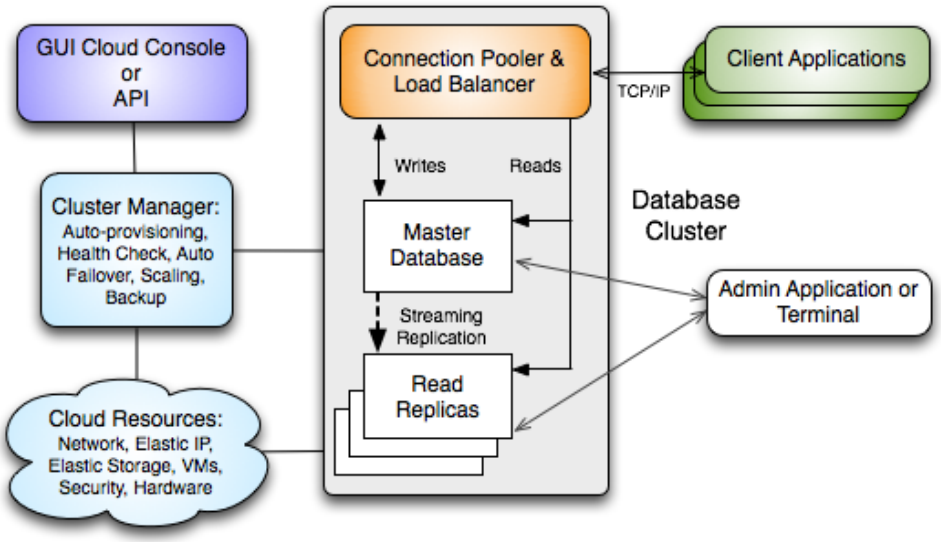


Figure 2.1 - An overview of the EDB Ark architecture.

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.

Copyright © 2011 - 2017 EnterpriseDB Corporation. All rights reserved.

EDB Ark installs Postgres on each replica node in a read-only hot-standby role that automatically duplicates all data found on the master node, and all changes made to that data. In hot-standby mode, the data is available to service user queries providing read scalability to the cluster (see Figure 2.2). In addition, any schema changes made to the master are also replicated to the replica nodes, making development and deployment of application changes easy and seamless without interruption to normal operations.

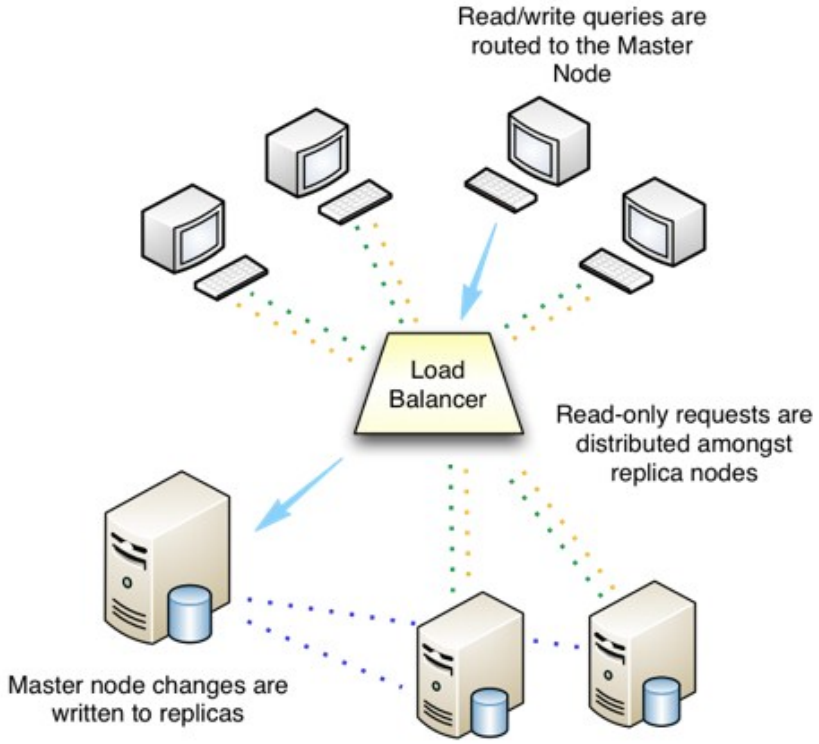


Figure 2.2 - EDB Ark performs automatic load balancing.

Replicas provide balanced user support as needed - if any instance in the cluster goes offline, the cluster's load is re-balanced among the remaining servers while the instance is automatically replaced.

When used in the default healing configuration, in the event of a failure of the master node, an existing replica is used to replace the failed master node. While the replica nodes are standing by, they are read-only resources, load balancing client queries without a risk of compromising data integrity.

EDB Ark automatically archives data at regular intervals; you can specify a convenient backup window and how many backups to retain when creating a database cluster. EDB Ark also offers backup on demand - simply click the `Backup` icon to save a copy of the instance. Automatic backups are retained according to your specifications; while 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.

Copyright © 2011 - 2017 EnterpriseDB Corporation. All rights reserved.

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.

Copyright © 2011 - 2017 EnterpriseDB Corporation. All rights reserved.

2.2 Supported Platforms

The EDB Ark management console runs on the following browser versions (or newer):

- Mozilla Firefox 18
- Mozilla Firefox 17 ESR, 24 ESR, 31 ESR
- Internet Explorer 8
- Safari 6
- Opera 16
- Google Chrome 23

EDB Ark console is supported on the following OpenStack releases:

- Community OpenStack Mitaka

EDB Ark provisions cluster instances on Amazon public clouds on the following 64-bit Linux systems:

- CentOS 7.x and 6.x

Copyright © 2011 - 2017 EnterpriseDB Corporation. All rights reserved.

2.3 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.

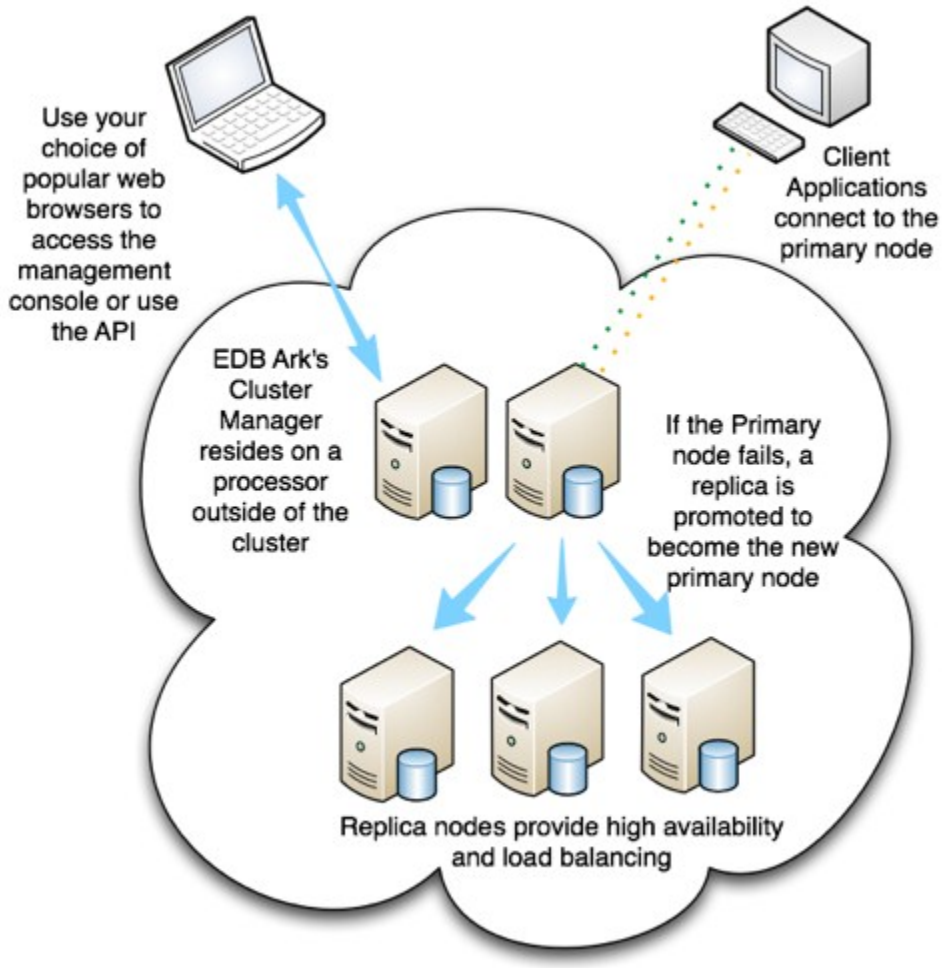


Figure 2.3 - Using EDB Ark in a Cloud.

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.

Copyright © 2011 - 2017 EnterpriseDB Corporation. All rights reserved.

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.

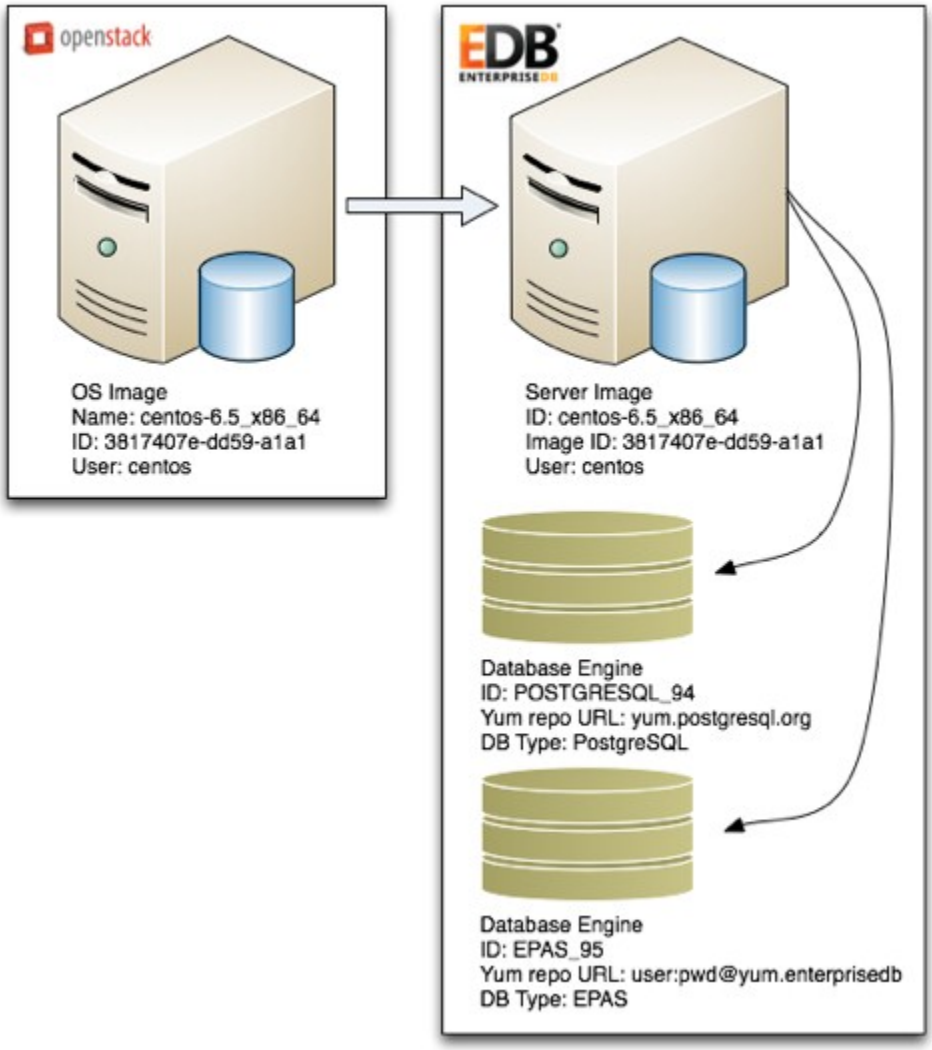


Figure 2.4 – Using an OpenStack image with EDB Ark.

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.

Copyright © 2011 - 2017 EnterpriseDB Corporation. All rights reserved.

2.4 Using Ark on an Amazon AWS Virtual Private Cloud

EDB Ark can create and manage cloud clusters that reside on Amazon-hosted virtual private networks. A virtual private cloud (VPC) is similar in structure to a traditional network, but provides the scalability and ease of maintenance offered by cloud computing.

A VPC is an isolated network with a unique IP address range and subnet address (or addresses). When you use the Ark console to create a cloud instance within a VPC, you specify the ID of the private cloud, and Ark assigns the new instance an IP address from within your private network.

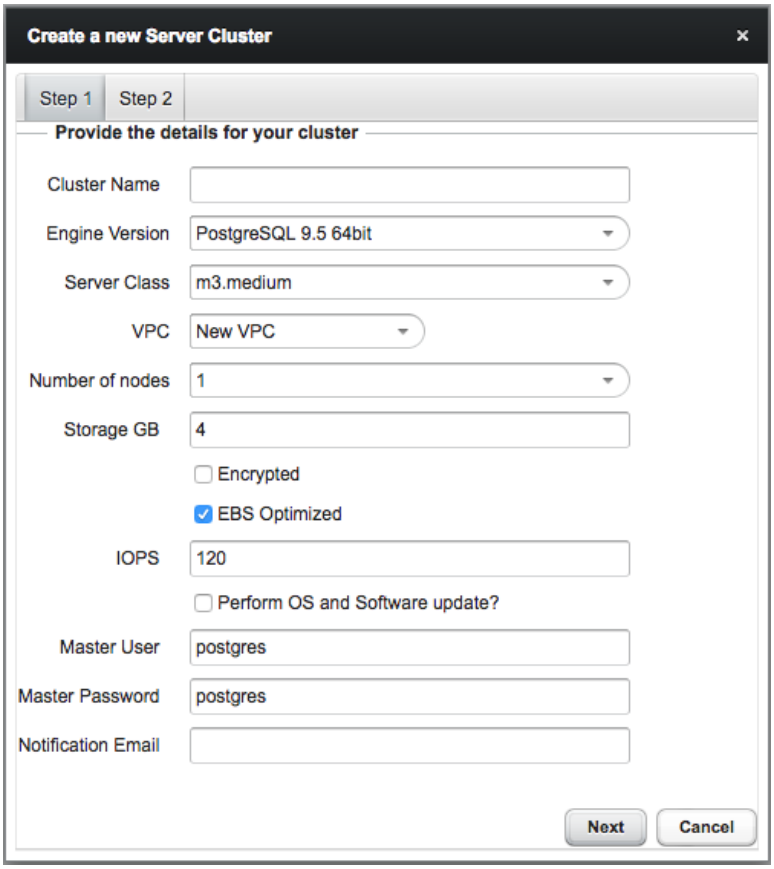


Figure 2.5 - Creating a new Ark cluster.

To create a new cluster that resides on a VPC, log into the Ark console and click the `Launch DB Cluster` button. When the `Create a new Server Cluster` dialog opens (as shown in Figure 2.5), provide details about the cluster configuration. Use the `VPC` drop-down menu to select an existing VPC, or choose `New VPC` to create a new virtual private cloud into which the cluster will be deployed. EDB Ark will create the new instance on a virtual machine in the specified VPC network.

Copyright © 2011 - 2017 EnterpriseDB Corporation. All rights reserved.

2.5 Prerequisites

2.5.1 Disable Pop-Up Blockers

Some features of the Ark Administrative console will not work properly when pop-up blocker (or Ad blocker) software is enabled. To take full advantage of console features, you should disable pop-up blocker software from restricting pop-ups from the URL/s used by the Ark console or Ark clusters.

2.5.2 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 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.

2.5.3 Using EDB Ark on an OpenStack Mitaka Host

By default, OpenStack Mitaka enables the Keystone identity service version 3.0 API; version 3.0 is not supported by EDB Ark. Before using EDB Ark on an OpenStack Mitaka host, you must enable the Keystone identity service version 2.0 API. Use the following process to enable the version 2.0 API for your domain:

1. Use the OpenStack command line to retrieve the list of OpenStack domains:

```
(openstack) domain list
Password:
+----------------------------------+---------+---------+----------------+
| ID                               | Name    | Enabled | Description    |
+----------------------------------+---------+---------+----------------+
| b77a32b08b2345faa81f5fa706369b1d | default | True    | Default Domain |
+----------------------------------+---------+---------+----------------+
```

2. Connect to the Keystone server(s) and edit the `keystone.conf` file; by default, the file is located in `/etc/keystone/keystone.conf`.

Copyright © 2011 - 2017 EnterpriseDB Corporation. All rights reserved.

3. Modify the `[identity]` section of the keystone.conf file, setting the `default_domain_id` property to the ID of the chosen domain. For example:

```
default_domain_id = b77a32b08b2345faa81f5fa706369b1d
```

4. Restart the Keystone services. On a Community Openstack installation that has been configured on CentOS using the instructions in the community installation guide, you must also restart the Apache HTTPD server under which Keystone runs as a WSGI service. For example, on a CentOS 7.x host, use the command:

```
systemctl restart httpd
```

   If your installation requires you to restart the Keystone service directly, you can use the command:

```
systemctl restart openstack-keystone
```

2.5.4 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 2.6).

Figure 2.6 – 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.

Copyright © 2011 - 2017 EnterpriseDB Corporation. All rights reserved.

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.

2.5.5 Creating the EDB Ark Service Account on OpenStack

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 and platform-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.

Copyright © 2011 - 2017 EnterpriseDB Corporation. All rights reserved.

2.5.6 Creating the Amazon AWS Service User and Service Role

Before configuring the Ark console on an Amazon host and creating users, you must create an Amazon service user and service role. Ark uses the service role when performing Ark management functions (such as console backups). The Ark console uses the service role credentials (the cross account keys) to assume the IAM roles assigned to Ark users. This enables Ark to securely manage AWS resources.

When configuring the Ark console, you are required to provide details about the AWS service user and the service role in the `ppcd.properties` file. Specify:

- the Amazon Role ARN (resource name) that will be used by the Ark service in the `aws.service.account.rolearn` property.
- the Amazon external ID that will be used by the Ark service user (`ppcd`) in the `aws.service.account.externalid` property.
- the `AWS_ACCESS_KEY_ID` associated with the AWS role used for account administration in `aws.cross.account.accesskey` property.
- the `AWS_SECRET_ACCESS_KEY` associated with the AWS role used for account administration in `aws.cross.account.secretkey` property.

Copyright © 2011 - 2017 EnterpriseDB Corporation. All rights reserved.

2.5.6.1 Creating the AWS Service User

To create the Ark console's service user account, connect to the Amazon AWS management console, and navigate to the `Users` dashboard; select the `Add user` button to open the `Add user` dialog (shown in Figure 2.7).

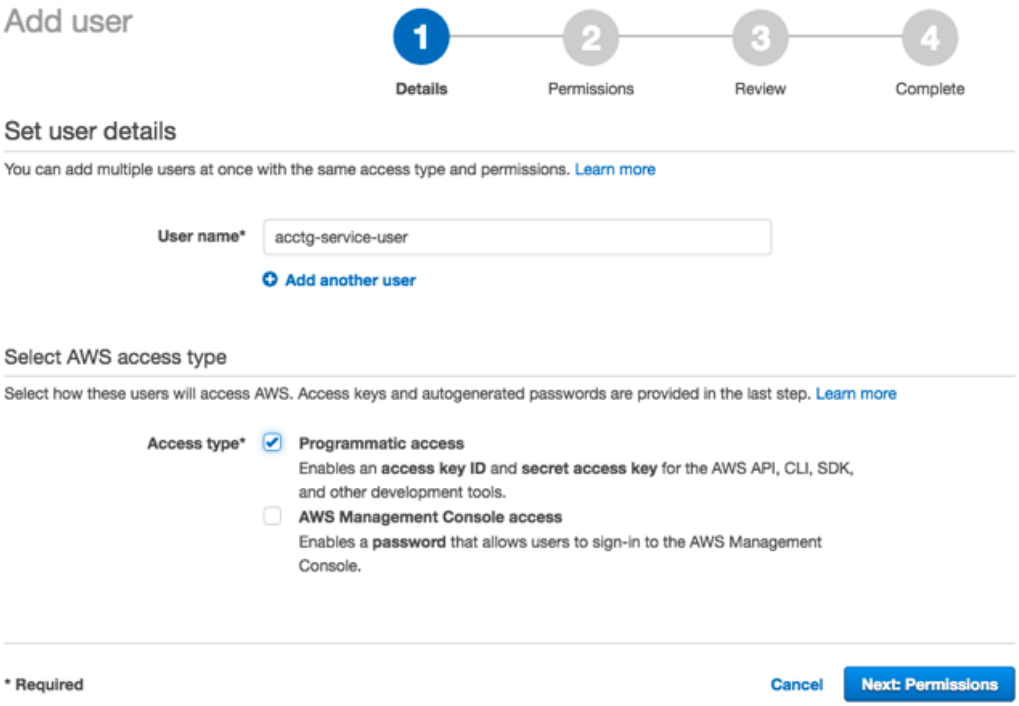


Figure 2.7 - The Add user dialog.

On the `Add user` dialog:

- Provide a name for the service user account in the `User name` field.
- Check the box to the left of `Programmatic access`.

Click `Next: Permissions` to continue.

When the `Permissions` dialog opens, click the button labeled `Attach existing policies directly`, then click the `Create policy` button. When the `Create Policy` dialog opens, click the button to the right of `Create Your Own Policy`.

The browser will open another tab, allowing you to define a custom policy (see Figure 2.8).

Copyright © 2011 - 2017 EnterpriseDB Corporation. All rights reserved.

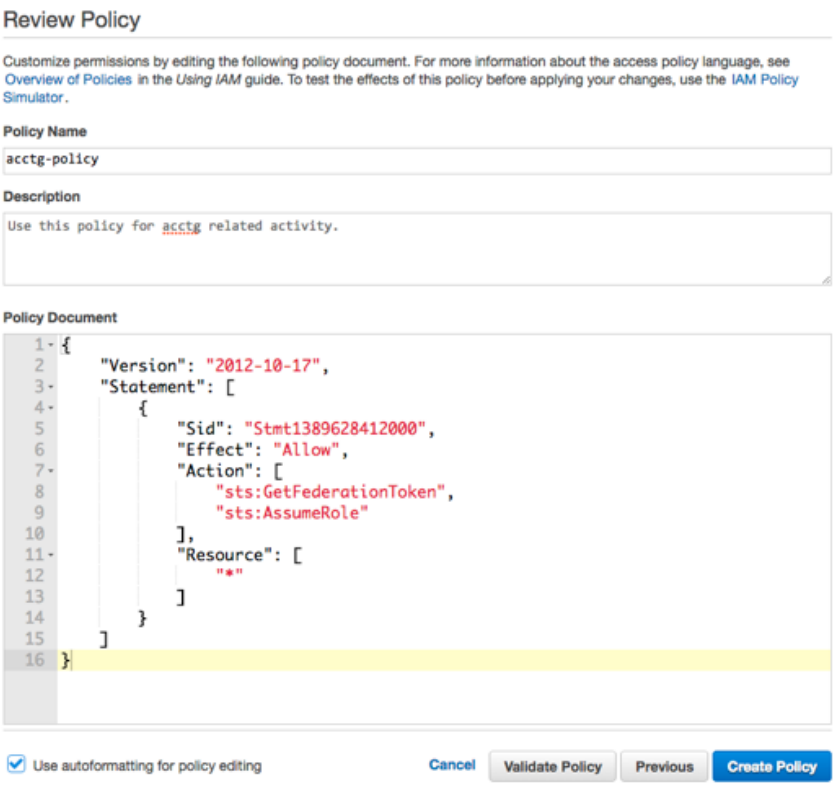


Figure 2.8 - The Review Policy dialog.

On the `Review Policy` dialog:

- Provide a name for the policy in the `Policy Name` field.
- Provide a description of the policy in the `Description` field.
- Provide the text that defines the policy in the `Policy Document` field. You can use the policy provided in *Reference - AWS Service User Security Policy*.

Click `Create Policy` to continue.

Then, return to the `Add user` dialog, and click the `Refresh` button above the list of policies. Select the new policy from the list (see Figure 2.9), and click `Next`.

Copyright © 2011 - 2017 EnterpriseDB Corporation. All rights reserved.

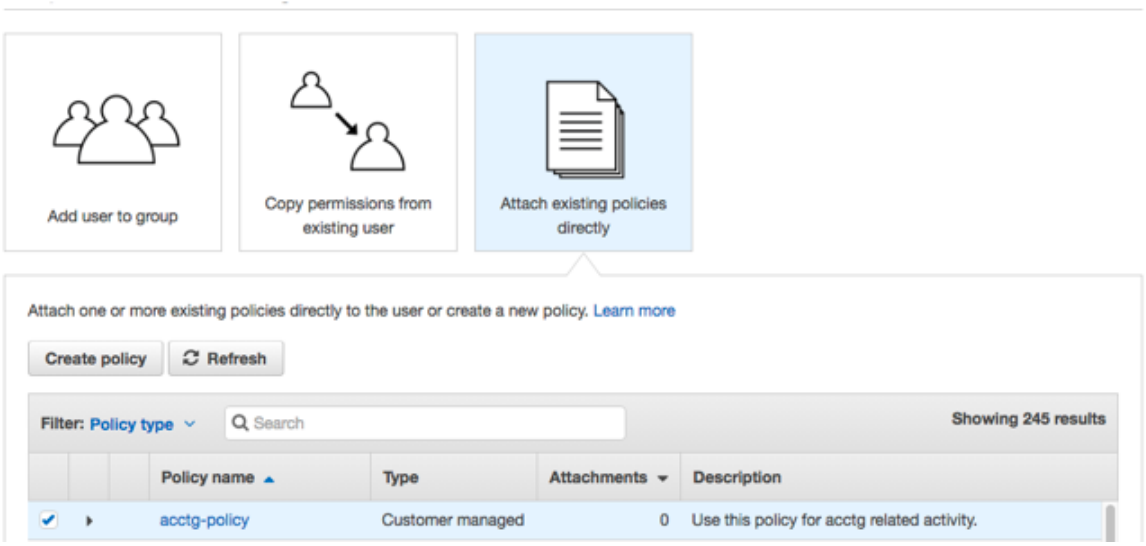


Figure 2.9 – Review the Add user dialog.

Confirm that the correct policy has been attached, and click `Create user`. The AWS console will confirm that the user has been added successfully. Click `Show` to display the `Secret access key` value (see Figure 2.10).

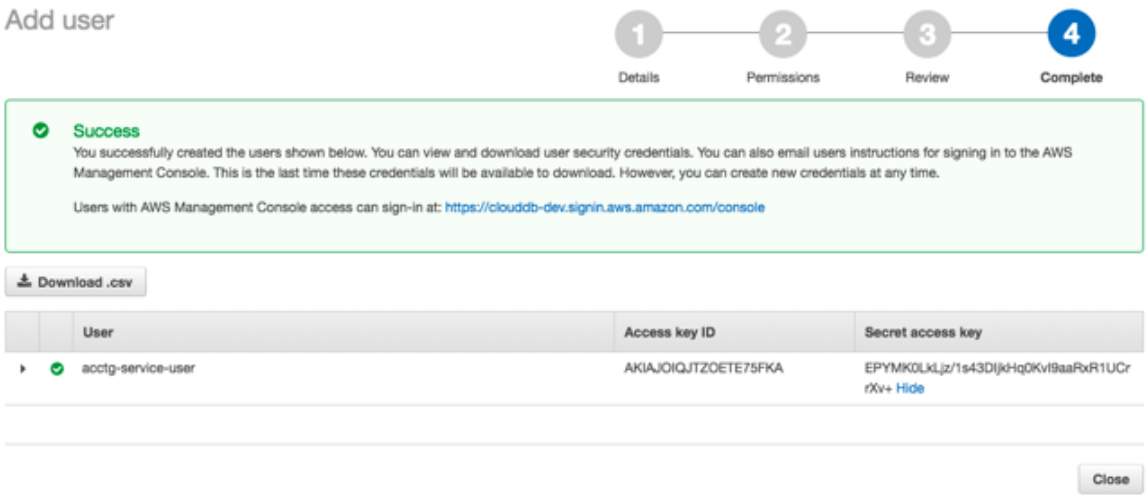


Figure 2.10 – Retrieving Access key information.

Copy the access key values displayed on the console; you must provide the values in the `ppcd.properties` file when configuring your Ark console:

- Provide the `Access key id` in the `aws.cross.account.accesskey` parameter.
- Provide the `Secret access key` in the `aws.cross.account.secretkey` parameter.

Copyright © 2011 - 2017 EnterpriseDB Corporation. All rights reserved.

2.5.6.2 Creating the AWS Service Role

After creating the service user, you must create a service role. To define a service role, connect to the Amazon management console, and navigate to the `Identity and Access Management` Dashboard (see Figure 2.11).

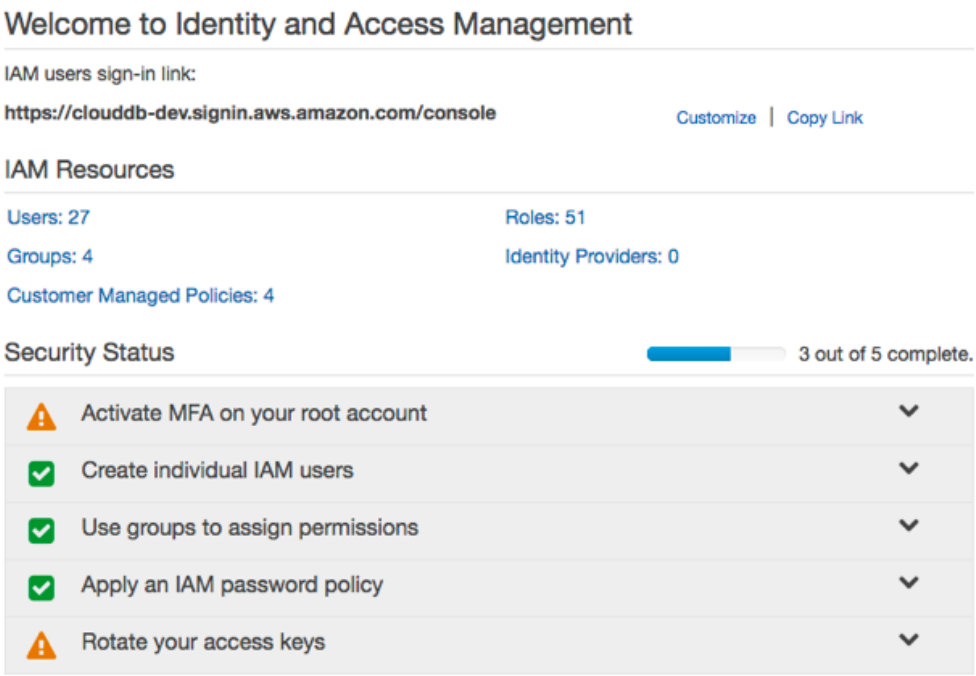


Figure 2.11 - The Amazon IAM Dashboard.

Navigate to the `Roles` page, and click the `Create New Role` button.

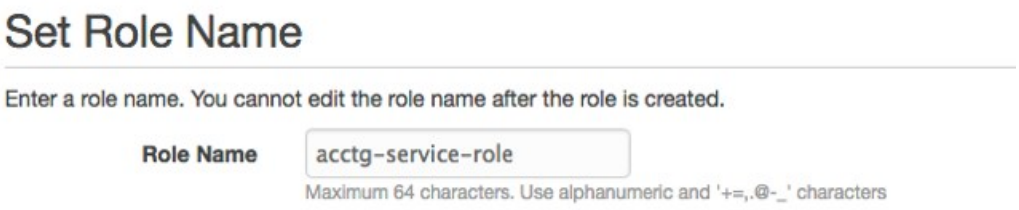


Figure 2.12 - Provide a role name.

When the `Create Role` dialog opens (shown in Figure 2.12), specify a name for the new role and click `Next Step` to specify a role type.

Copyright © 2011 - 2017 EnterpriseDB Corporation. All rights reserved.

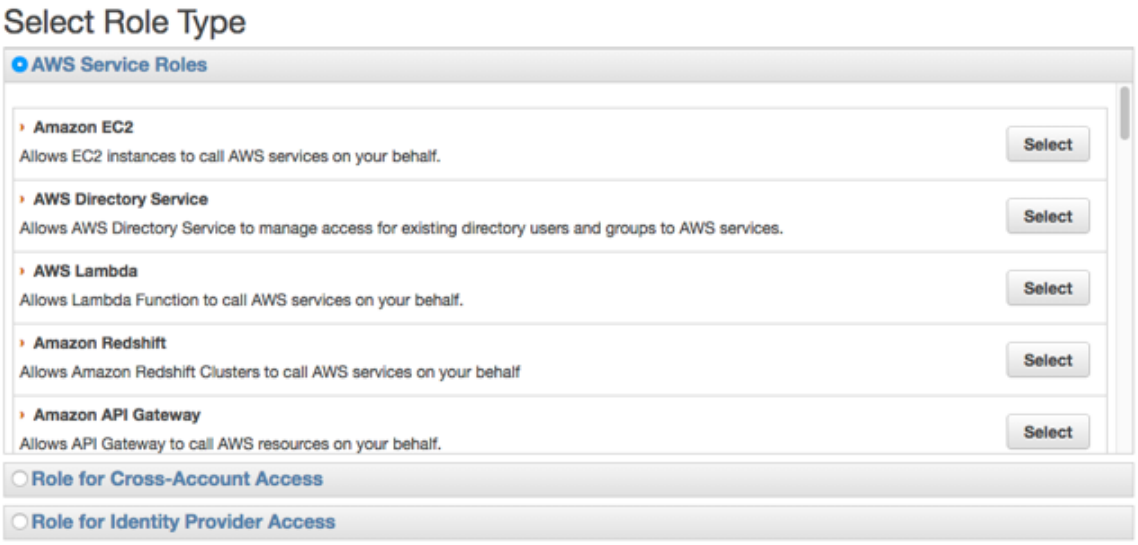


Figure 2.13 - Specify that the role allows EC2 instances to call AWS services.

Select the `AWS Service Roles` radio button (shown in Figure 2.13), and then the `Select` button to the right of `Amazon EC2` to continue to the `Attach Policy` dialog.

Copyright © 2011 - 2017 EnterpriseDB Corporation. All rights reserved.

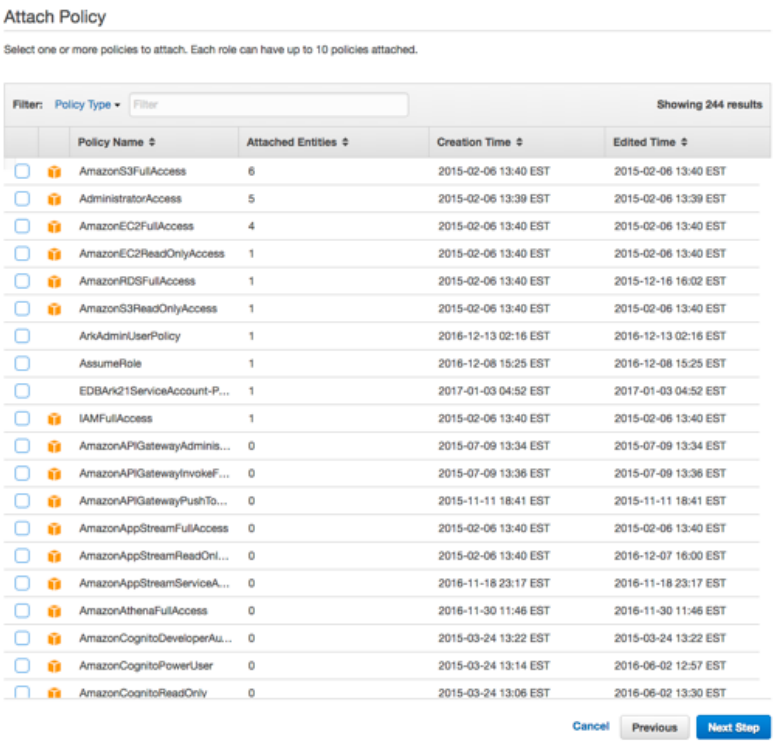


Figure 2.14 – The Attach Policy dialog.

When the `Attach Policy` dialog (shown in Figure 2.14) opens, do not select a policy; instead, click `Next Step` to continue to the `Review` dialog.

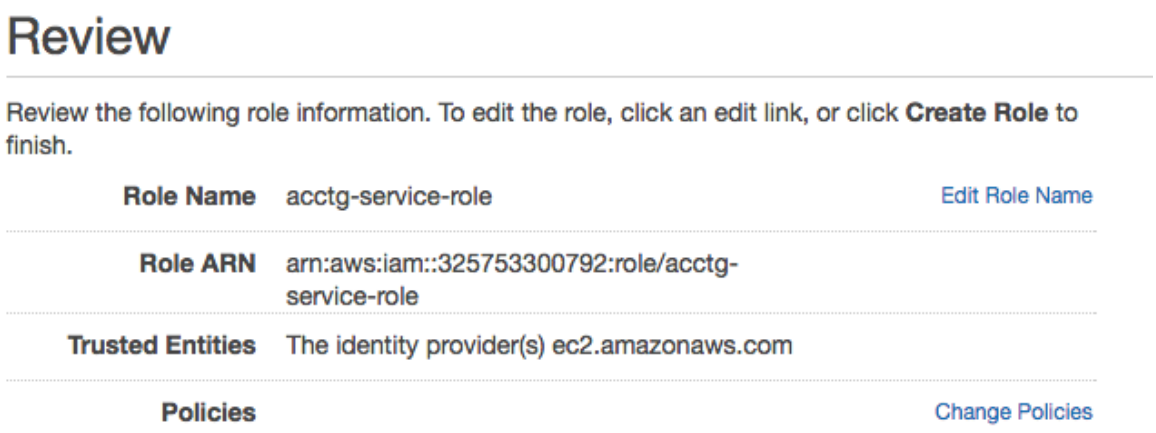


Figure 2.15 - Review the role information.

When the `Review` dialog opens (see Figure 2.15), review the information displayed, and then click `Create Role` to instruct the AWS management console to create the described role.

Copyright © 2011 - 2017 EnterpriseDB Corporation. All rights reserved.

Figure 2.16 - The new role is displayed on the Roles page.

The role will be displayed in the role list on the Amazon IAM `Roles` page (see Figure 2.16). You can click the role name to display detailed information about the role. Please note that the `Summary` tab will display a `Role ARN`, but the ARN will not be enabled until the security policy and trust policy are updated.

After completing the `Create Role` wizard, you must modify the inline security policy and trust relationship to allow Ark to use the role. Highlight the role name, open the `Inline Policies` menu, and select `click here` to add a new policy.

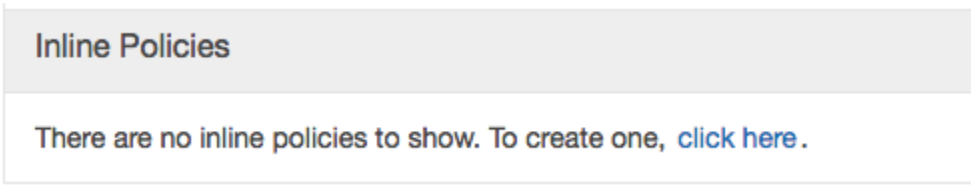


Figure 2.17 - The Inline Policies menu.

When the `Set Permissions` dialog opens, select the `Custom Policy` radio button, and then click the `Select` button (see Figure 2.18).

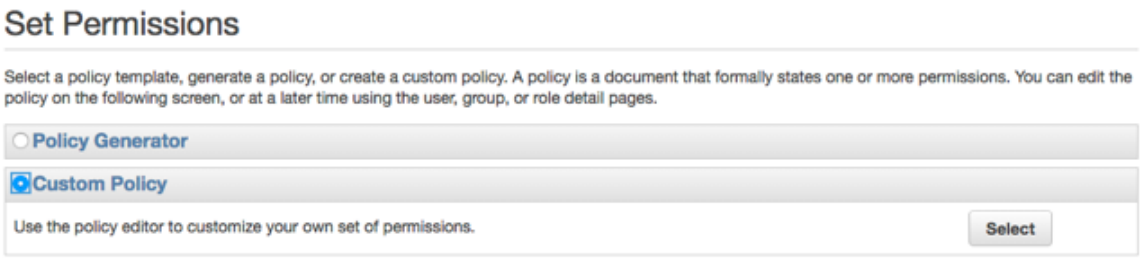


Figure 2.18 - Add a Custom Policy.

Copyright © 2011 - 2017 EnterpriseDB Corporation. All rights reserved.

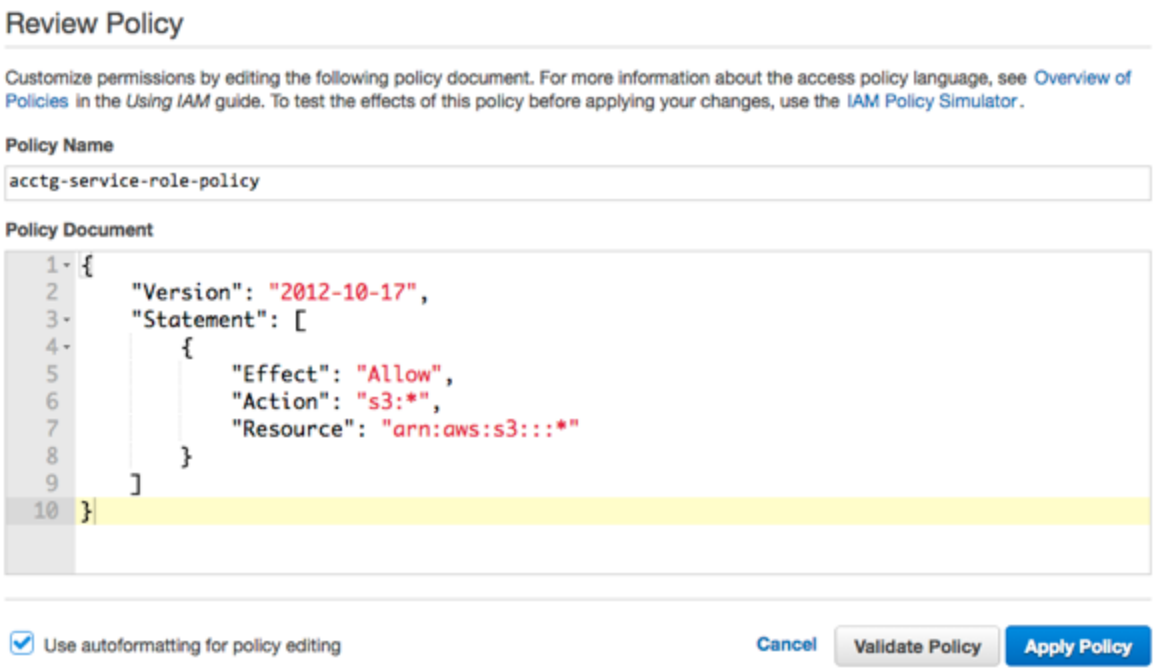


Figure 2.19 - Provide the policy name and contents.

Use the fields on the `Set Permissions` dialog (Figure 2.19) to define the security policy:

- Provide a name for the security policy in the `Policy Name` field.
- Copy the security policy text into the `Policy Document` field. For a sample security policy that you can use when creating the service role, please see *Reference – AWS Service Role Security Policy and Trust Relationship*.

After providing security policy information, click `Apply Policy` to return to the Role information page. Then, select the `Edit Trust Relationship` button (located in the `Trust Relationships` section) to display the `Policy Document` (see Figure 2.20).

Copyright © 2011 - 2017 EnterpriseDB Corporation. All rights reserved.

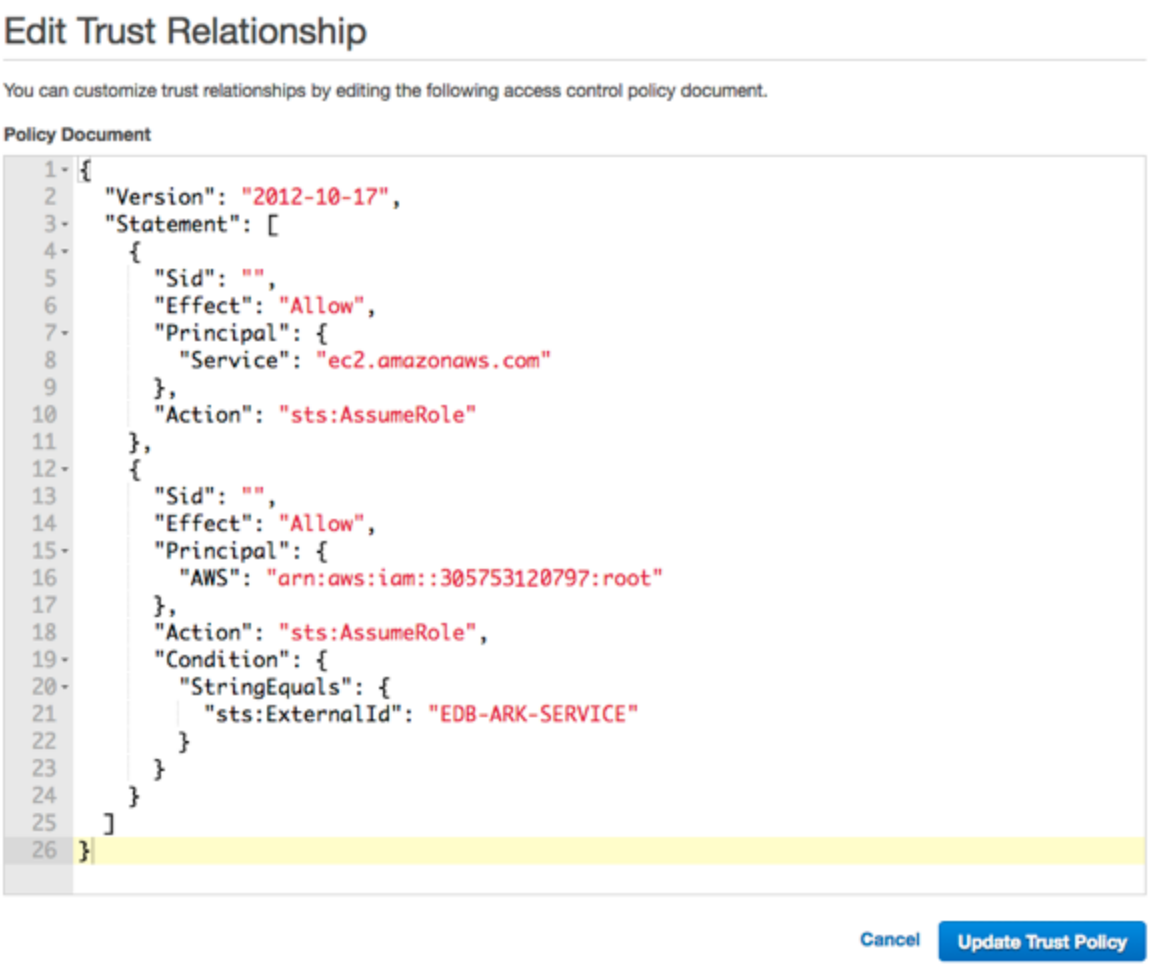


Figure 2.21 - The Policy Document.

Replace the displayed content of the policy document with the content of the security policy included in *Reference – AWS Service Role Security Policy and Trust Relationship*. Click the `Update Trust Policy` button to finish and close the `Edit Trust Relationship` dialog .

Copyright © 2011 - 2017 EnterpriseDB Corporation. All rights reserved.

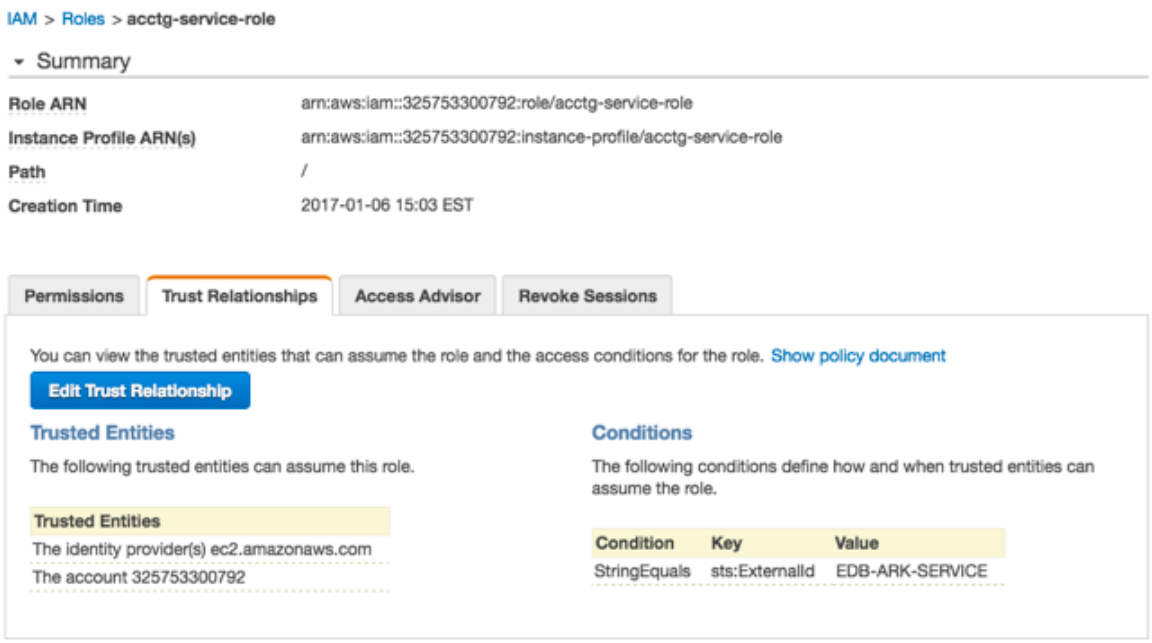


Figure 2.22 - The Summary tab of the Role detail panel.

The `Summary` dashboard will display values that you must provide in the `ppcd.properties` file when configuring your Ark console:

- The `Role ARN` associated with the service role must be provided in the `aws.service.account.rolearn` parameter.
- The external ID associated with the service role must be provided in the `aws.service.account.externalid` parameter. In the example shown, the external id is `EDB-ARK-SERVICE`; you can find this value under the `Conditions` section of the `Trust Relationships` tab.

Copyright © 2011 - 2017 EnterpriseDB Corporation. All rights reserved.

3 Installing the EDB Ark Console

If your cluster resides on an Amazon public cloud, please see Section 3.1 for detailed console installation information.

If your cluster uses an OpenStack host, please see Section 3.2 for detailed console installation information.

3.1 Installing EDB Ark for Amazon AWS

The EDB Ark console is distributed through the Amazon AWS Marketplace in an Amazon machine instance. To install the Ark console on your Amazon instance, you will need to:

1. Connect to your Amazon AWS Marketplace Account.

2. Locate the AMI that contains the Ark console, and launch the EDB Ark instance.

3. Use ssh to connect to the Ark host, and update the `ppcd.properties` configuration file. For more information, see Section 3.1.2.

4. Deploy the Ark console. For more information, see Section 3.1.3.

5. Create an Amazon role and register an administrative user. For more information, see Section 3.1.4.

Copyright © 2011 - 2017 EnterpriseDB Corporation. All rights reserved.

3.1.1 Launching an Amazon AWS Marketplace AMI

Before launching an AMI into an Amazon VPC, you must ensure that the VPC has access to an Internet Gateway. If your VPC does not have access to an Internet Gateway, you can use the Amazon management console to create an Internet Gateway and associate it with your VPC. Please note: if you are using EC2-Classic networking, you do not need to provide an Internet Gateway.

For detailed information about creating and using an Internet Gateway, see the Amazon documentation at:

http://docs.aws.amazon.com/AmazonVPC/latest/UserGuide/VPC_Internet_Gateway.html

To create an Amazon Machine Instance (AMI) that contains a running copy of GlassFish, the Ark console, and the Ark console's backing database, connect to your Amazon AWS Marketplace Account and locate the AMI that contains the Ark console. Navigate through the introductory page for the AMI, selecting AWS service options that are appropriate to your application, and agreeing to the Terms and Conditions. When you agree to the Terms and Conditions, Amazon will process the subscription.

After you subscribe, Amazon will forward an email to the address associated with your user account that includes launch instructions for the AMI. For additional information about launching software from the AWS Marketplace, please refer to the online resources for Amazon Marketplace:

http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/launching-instance.html

Please note that when configuring your security group (see Step 9 of the AWS documentation referenced above, and Step 6 of the launch process), the group must allow communication between the nodes of the cluster.

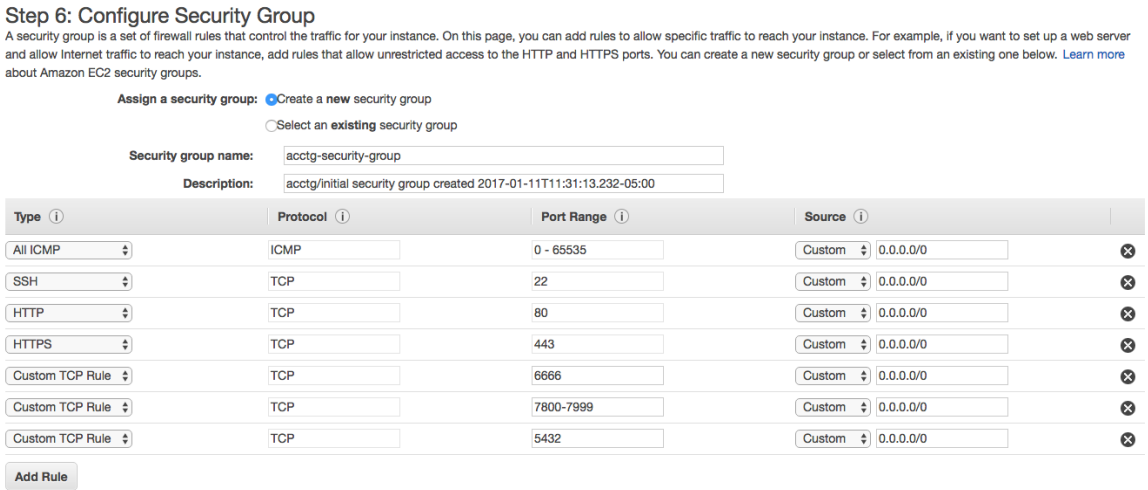


Figure 3.1 – Defining a Security Group when launching an AMI.

Copyright © 2011 - 2017 EnterpriseDB Corporation. All rights reserved.

When defining the security group, include 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 7999	CIDR	0.0.0.0/0

The CIDR addresses specified in the rules for SSH, HTTP, HTTPS, and 5432 can be customized to restrict access to a limited set of users. The CIDR addresses specified for port 6666 and ports 7800 through 7999 must specify a value of `0.0.0.0/0`.

The `Custom TCP` rule that opens ports `7800` through `7999` provides enough ports for 200 cluster connections; the upper limit of the port range can be extended if more than 200 clusters are required.

3.1.2 Configuring the Installation

When the launch of your instance completes, you can review the system log to confirm the status of the GlassFish application server and the backing PostgreSQL database. To review the system log, connect to the Amazon Management Console and navigate to the `Instances` dashboard. Highlight the instance name in the list and open the `Actions` drop-down menu; navigate through the `Instance Settings` menu, selecting `Get System Log`.

After confirming that the services are running, you can configure the installation. Use the identity associated with the Amazon AMI and the SSH key associated with the instance on which the console will reside to SSH to the console host:

```
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 IP address of the Ark console.

Copyright © 2011 - 2017 EnterpriseDB Corporation. All rights reserved.

Setting the Console Time Zone

After connecting with SSH, assume `root` privileges, and use the following commands to set the console time zone.

```
# rm /etc/localtime
# ln -s /usr/share/zoneinfo/time_zone /etc/localtime
# rm –f /etc/timezone
# ln –s /usr/share/zoneinfo/time_zone /etc/timezone
```

Where *`time_zone`* specifies the time zone identifier that the console will use. To discover the available time zones for your system, you can use the command:

```
ls -l /usr/share/zoneinfo/
```

Then, restart the console's Postgres server, and then the GlassFish server:

```
# /etc/init.d/postgresql-9.6 restart
# su – ppcd
$ asadmin restart-domain
```

Then, use your choice of editor to modify the `ppcd.properties` file.

3.1.2.1 Configuring the ppcd.properties File on an Amazon Host

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.

Likewise, parameters that are prefaced with `aws` have values that correspond to values specified on the Amazon AWS management console. The value specified on the Amazon AWS Management console 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

Copyright © 2011 - 2017 EnterpriseDB Corporation. All rights reserved.

commented out; if you uncomment them, 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= 0f42d1934a1a19f3d25d6288f2a3272c6143fc5d
# DB name to connect to
# console.db.name=postgres
```

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

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 Ark user account (`ppcd`).

```
# 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. Please note that you must create the directory specified. The Ark user account (`ppcd`) 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
```

On an Amazon hosted console, you can use the `console.db.backup.container` and `console.db.backup.folder` parameters to specify the name of a container (an Amazon S3 bucket) in which console backups will be stored, and a console-specific folder name. If no value is specified for `console.db.backup.folder`, the value will default to `default`.

```
# Optional bucket name in which to store console backups
# console.db.backup.container=

# Unique name for the console backup folder that identifies this
# console, i.e. 'dev.console'. Default name is 'default'
# console.db.backup.folder=
```

Copyright © 2011 - 2017 EnterpriseDB Corporation. All rights reserved.

Please note: Your AWS S3 backup container name must be unique when compared to the names of *all* other AWS containers. Including account specific information in the container identifier may help you create a unique name; for example:

```
account-name.console.db.backup.container
```

For more information about forming a bucket name, please consult the Amazon documentation at:

http://docs.aws.amazon.com/AmazonS3/latest/dev/BucketRestrictions.html

Please note: backups are first created in the location specified in `console.db.backup.dir` before being copied to the container specified in `console.db.backup.container`. You must provide values for both parameters.

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

Copyright © 2011 - 2017 EnterpriseDB Corporation. All rights reserved.

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
```

Please note: If you are using an AWS S3 bucket, your bucket name must be unique when compared to the names of *all* other AWS buckets. Including account specific information in the bucket identifier may help you create a unique name; for example:

account-name.wal.archive.container

For more information about forming a bucket name, please consult the Amazon documentation at:

http://docs.aws.amazon.com/AmazonS3/latest/dev/BucketRestrictions.html

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

The parameters listed in the `OpenStack specific properties` section will not apply to those consoles that are installed on an Amazon AWS host.

```
# 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
```

Copyright © 2011 - 2017 EnterpriseDB Corporation. All rights reserved.

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 OpenStack 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
```

Amazon AWS specific properties

Use the `aws.service.account.rolearn` parameter to specify the Amazon RoleARN (resource name) that should be used by the Ark service user (`ppcd`) when performing management functions on behalf of Ark.

```
# the IAM role for the AWS service account
aws.service.account.rolearn=iam_role_arn
```

Use the `aws.service.account.externalid` parameter to specify the Amazon external ID that should be used by the Ark service user (`ppcd`).

```
# the external ID for the IAM role for the AWS service account
aws.service.account.externalid=iam_role_externalId
```

Use the `aws.region` parameter to specify the Amazon region in which Ark clusters will reside.

```
# the AWS region hosting your EDB Ark console (i.e. us-east-1)
aws.region=region_name
```

Use the `aws.cross.account.accesskey` parameter to specify the Amazon `AWS_ACCESS_KEY_ID` associated with the AWS role used for account administration.

```
# the AWS IAM cross account access key
aws.cross.account.accesskey=accesskeyid
```

Use the `aws.cross.account.secretkey` parameter to specify the Amazon `AWS_SECRET_ACCESS_KEY` associated with the AWS role used for account administration.

```
# the AWS IAM cross account secret key
aws.cross.account.secretkey=secretkeyid
```

Copyright © 2011 - 2017 EnterpriseDB Corporation. All rights reserved.

If your console uses an Amazon AWS backing host, you can use the `self.registration.enabled` parameter to instruct the Ark console to enable or disable self-registration for Ark users.

```
If self.registration.enabled is set to false, an administrative
user must register each Ark console user in the Ark
administrative console.
```

If self.registration.enabled is set to true, the Ark console login dialog will display a `Register` button. An unregistered console user can use the `Register` button to access a dialog that allows them to register their own user account, and access the console. To successfully register, the user must be able to access the AWS management console to retrieve a valid Amazon Role ARN that will be associated with their identity.

```
# Self registration enabled
self.registration.enabled=false
```

Display properties

Use the `console.dashboard.docs` and `console.dashboard.hot.topics` parameters to specify the source of the content that will be displayed on the Dashboard tab of the Ark console:

- If your cluster resides on a network with Internet access, set the parameters to `DEFAULT` to display content (alerts and documentation) from EnterpriseDB.
- If you would like the Dashboard tab to display alternate content, use the parameters to provide the URL of the content.
- If your cluster has limited access to the Internet, or if you wish to not display content on the Dashboard tab, leave the parameter values empty.

```
# these properties allow you to control the dashboard content.
# Legal values:
#       DEFAULT = load the default pages from enterprisedb.com
#       <unset> = don't load anything
#       <url>   = load alternate content at specified url
console.dashboard.docs=DEFAULT
console.dashboard.hot.topics=DEFAULT
```

Copyright © 2011 - 2017 EnterpriseDB Corporation. All rights reserved.

3.1.3 Deploying the Console

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

```
[root@ip-10-0-83-6 ~]# /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!
```

3.1.4 Creating an Amazon Role

After deploying the console, you must create an Amazon role with an associated security policy that will be applied to the Ark console user. You can use the same security policy for multiple users, or create additional Amazon roles with custom security policies for additional users. Each time you register a user, you will be prompted for a Role ARN. The Role ARN determines which security policy will be applied to that user.

To define an Amazon role, connect to the Amazon management console, and navigate to the `Identity and Access Management` dashboard (see Figure 3.2).

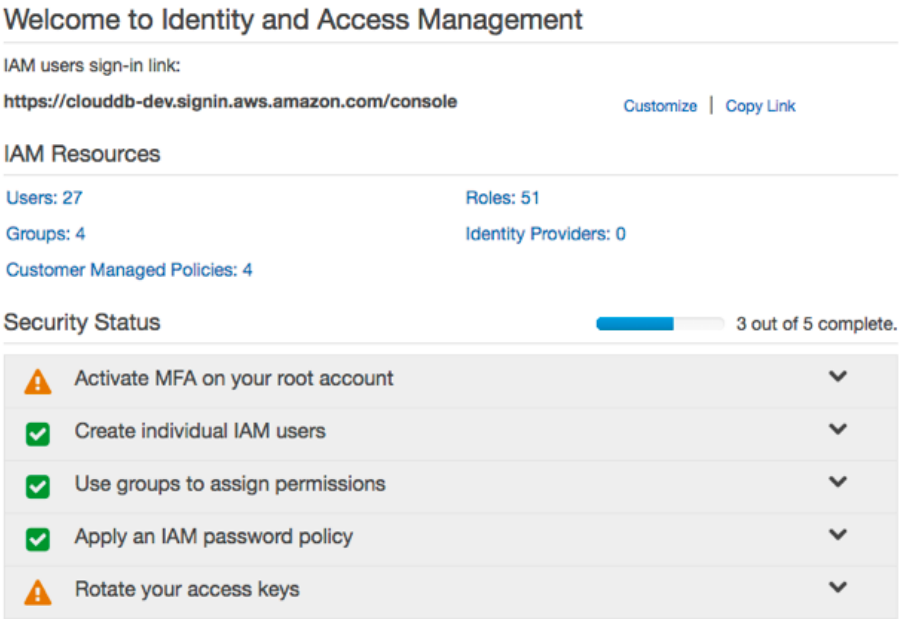


Figure 3.2 - The Amazon IAM Dashboard.

Copyright © 2011 - 2017 EnterpriseDB Corporation. All rights reserved.

Navigate to the `Roles` dashboard, and click the `Create New Role` button.

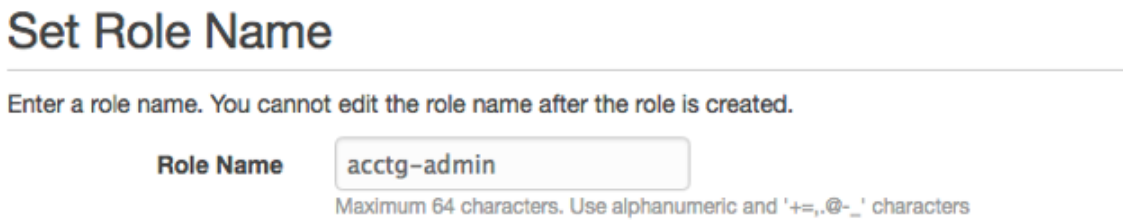


Figure 3.3 - Provide a role name.

When the `Set Role Name` dialog opens (shown in Figure 3.3), specify a name for the new role and click `Next Step` to specify a role type.

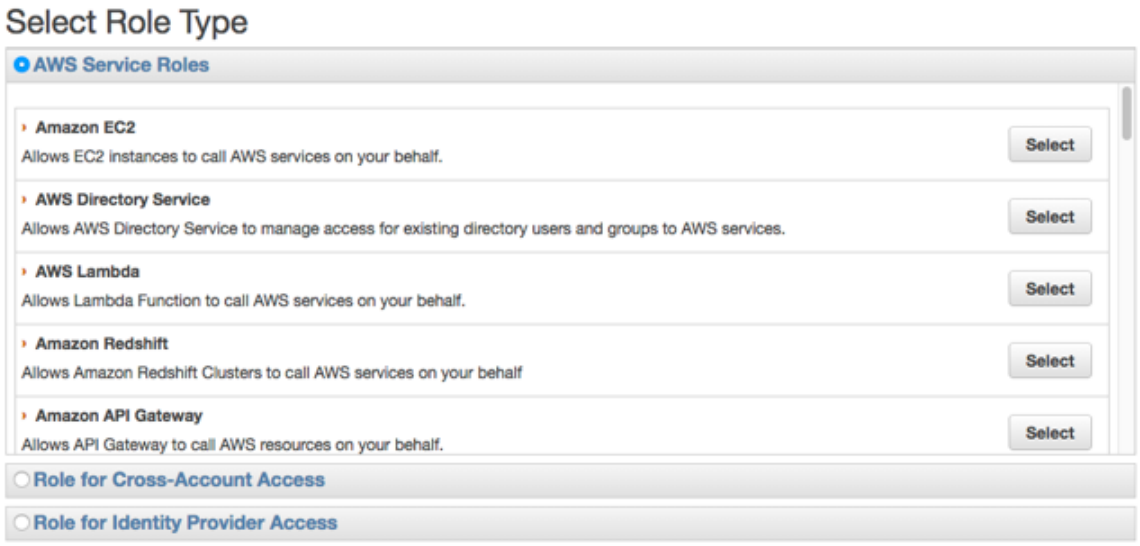


Figure 3.4 - Specify that the role allows EC2 instances to call AWS services.

On the `Select Role Type` dialog, select the `AWS Service Roles` radio button (shown in Figure 3.4), and then the `Select` button to the right of `Amazon EC2` to continue to the `Attach Policy` dialog.

Copyright © 2011 - 2017 EnterpriseDB Corporation. All rights reserved.

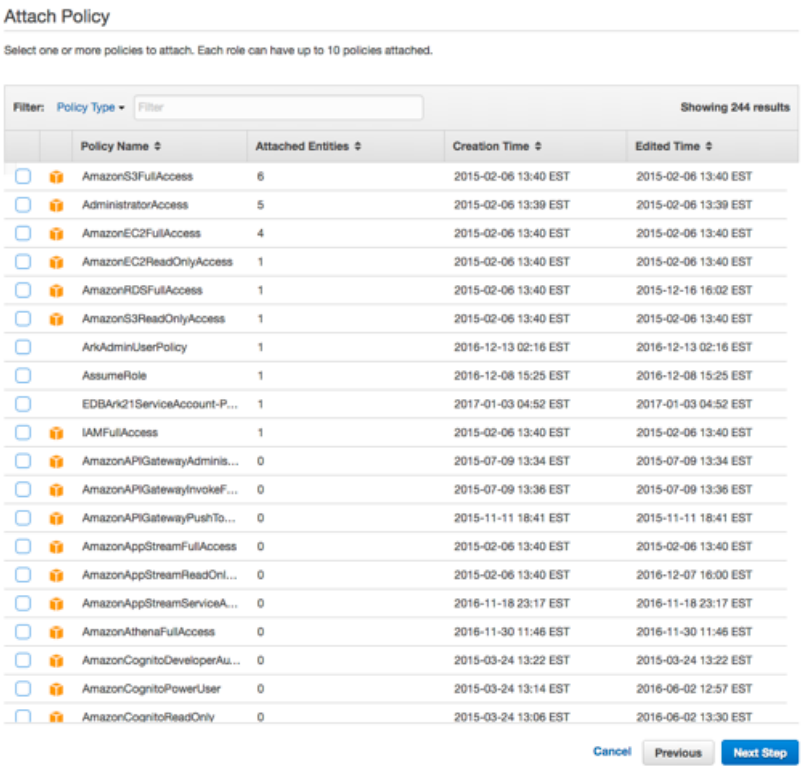


Figure 3.5 – The Attach Policy dialog.

When the `Attach Policy` dialog (shown in Figure 3.5) opens, do not specify a policy; instead, click `Next Step` to continue to the `Review` dialog.

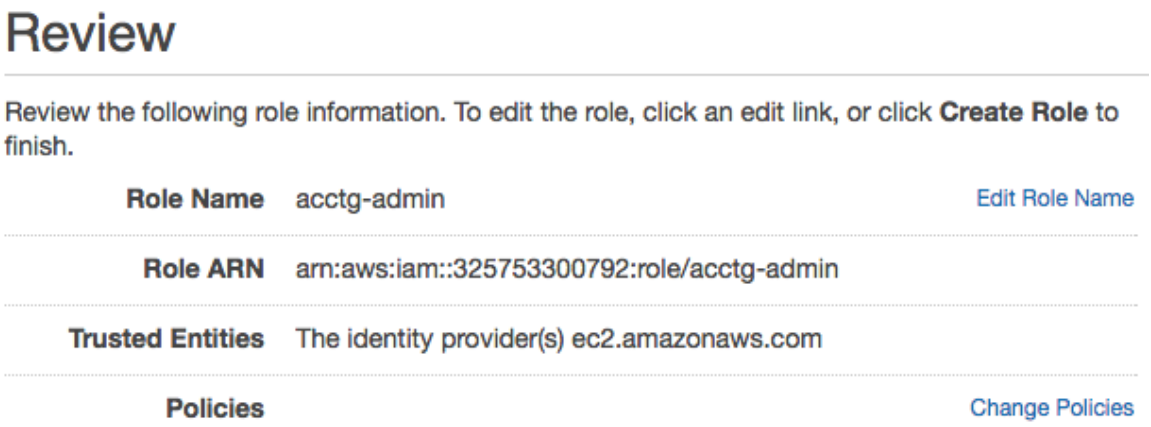


Figure 3.6 - Review the role information.

When the `Review` dialog opens (as shown in Figure 3.6), review the information displayed, and then click `Create Role` to instruct the AWS management console to create the described role.

Copyright © 2011 - 2017 EnterpriseDB Corporation. All rights reserved.

Figure 3.7 - The new role is displayed on the Roles page.

The role will be displayed in the role list on the Amazon IAM `Roles` page (see Figure 3.7). The `Summary` tab will display a `Role ARN`, but the ARN will not be enabled until the security policy and trust policy are updated.

After completing the `Create Role` wizard, you must modify the inline policy and trust relationship (defined by the security policy) to allow Ark to use the role. Highlight the role name; then open the `Inline Policies` menu and select `click here` to add a new policy.

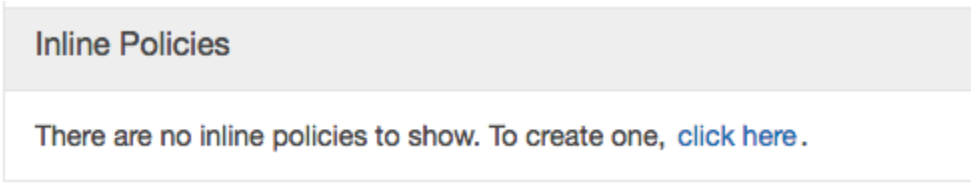


Figure 3.8 - The Inline Policies menu.

When the `Set Permissions` dialog opens, select the `Custom Policy` radio button, and then click the `Select` button (see Figure 3.9).

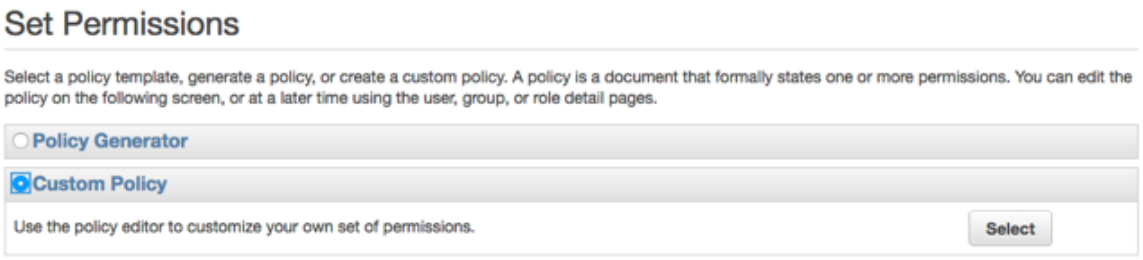


Figure 3.9 - Add a Custom Policy.

Copyright © 2011 - 2017 EnterpriseDB Corporation. All rights reserved.

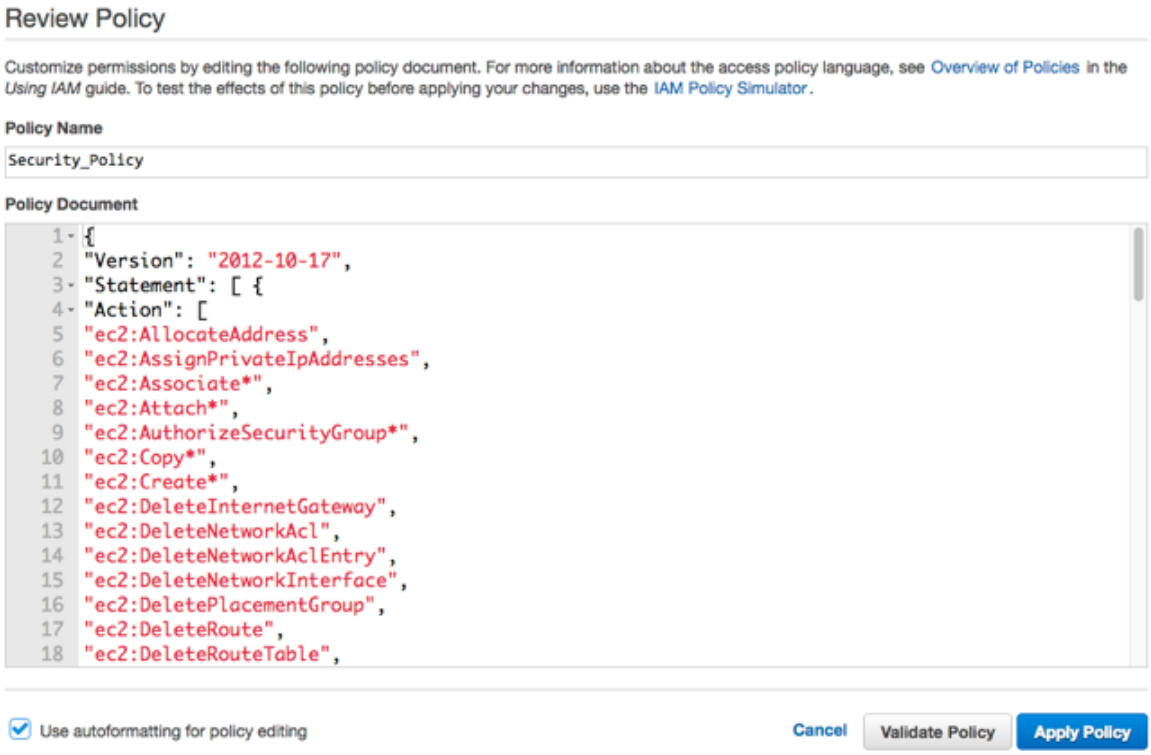


Figure 3.10 - Provide the policy name and contents.

Use the fields on the `Set Permissions` dialog (Figure 3.10) to define the security policy:

- Provide a name for the security policy in the `Policy Name` field.
- Copy the security policy text into the `Policy Document` field. The security policy required by Ark is available in Section 10.3, *AWS User Security Policy*.

After providing security policy information, click `Apply Policy` to return to the Role information page. Then, select the `Edit Trust Relationship` button (located in the `Trust Relationships` section) to display the `Policy Document` (see Figure 3.11).

Copyright © 2011 - 2017 EnterpriseDB Corporation. All rights reserved.

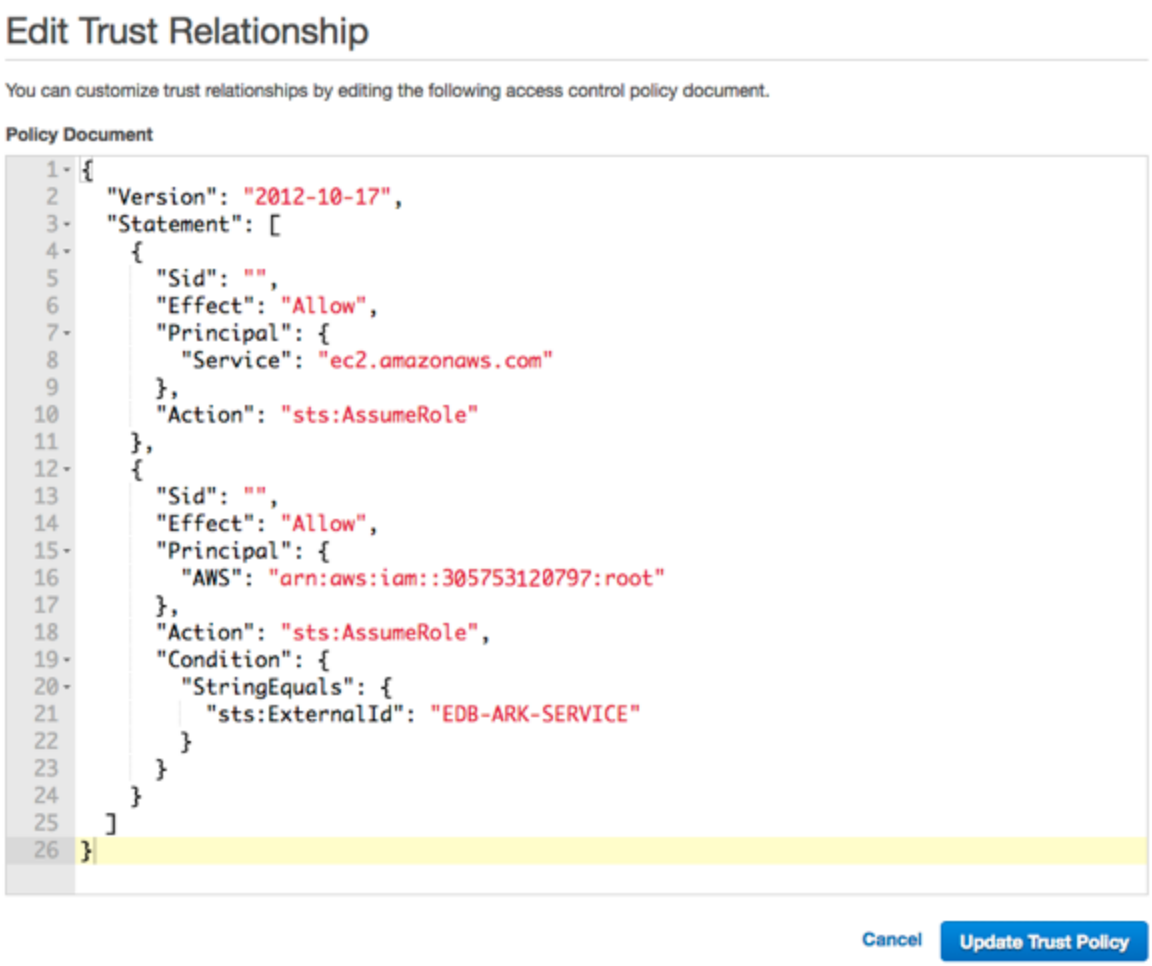


Figure 3.11 - The Policy Document.

Replace the displayed content of the policy document with the content of the file available in Section 10.4, *AWS User Trust Policy*.

`EDB-PPCD-CONSOLE` is a placeholder within the trust policy (see Figure 3.11). You must replace the placeholder with the `External ID` provided on the `Step 2` tab of the Ark console `New User Registration` dialog.

To retrieve the `External ID`, open another browser window and navigate to the `Log In` page of your Ark console. Click the `Register` button to open the `New User Registration` dialog (shown in Figure 3.12).

Copyright © 2011 - 2017 EnterpriseDB Corporation. All rights reserved.

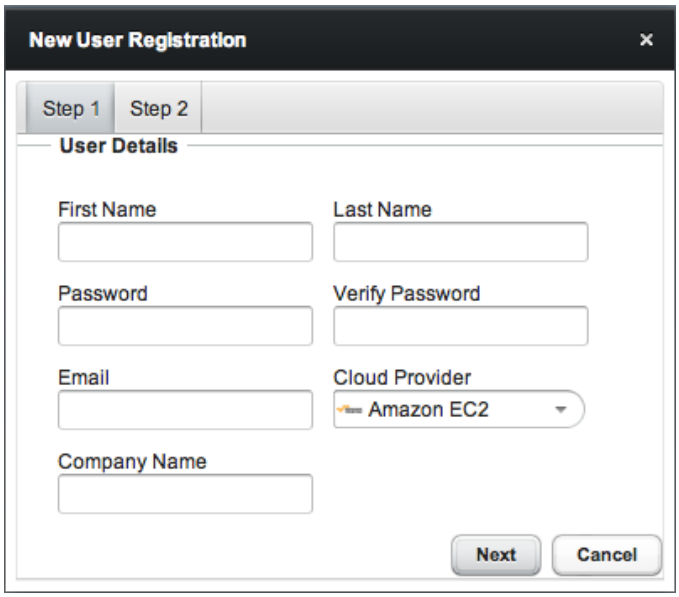


Figure 3.12 - The New User Registration dialog.

Enter user information in the `User Details` box located on the `Step 1` tab:

- Enter your first and last names in the `First Name` and `Last Name` fields.
- Enter a password that will be associated with the user account, and confirm the password in the `Password` and `Verify Password` fields.
- Provide an email address in the `Email` field; please note that the email address is used as the `Login` identity for the user.
- Use the drop-down listbox in the `Cloud Provider` field to select the host on which the cloud will reside.
- Enter the name of the company with which you are associated in the `Company Name` field.

When you've completed `Step 1`, click `Next` to open the `Step 2` tab (see Figure 3.14).

The `Step 2` tab of the `New User Registration` dialog will display a random `External ID` number. Copy the `External ID` from the `Step 2` dialog into the trust policy, replacing `EDB-PPCD-CONSOLE`. Please note that you must enclose the `External ID` in double-quotes (`"`). Click the `Update Trust Policy` button to save your edits and exit the dialog.

Copyright © 2011 - 2017 EnterpriseDB Corporation. All rights reserved.

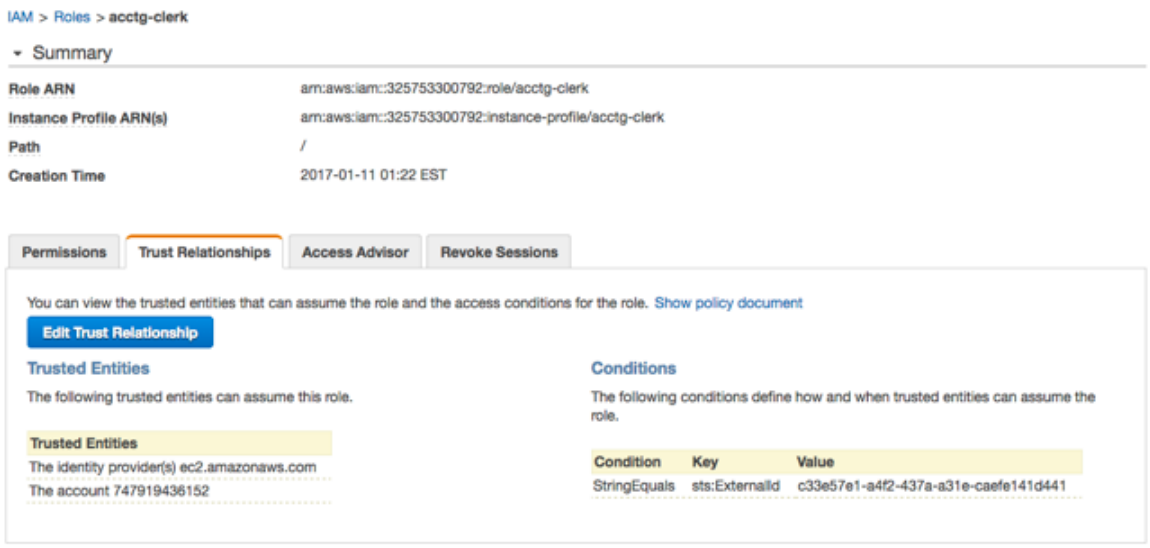


Figure 3.13 - The Summary tab of the Role detail panel.

Your Amazon IAM role ARN is displayed on the `IAM Roles` detail panel of the Amazon management console. Highlight a role name to display the assigned value on the `Summary` page (as shown in Figure 3.13).

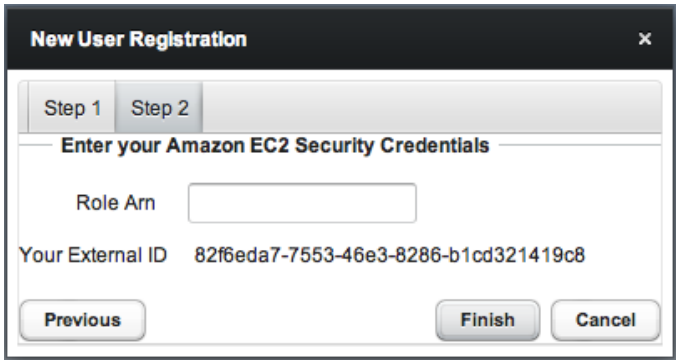


Figure 3.14 - Registering a user on an Amazon EC2 cloud.

Enter your Amazon IAM role ARN in the `Role Arn` field on the `Step 2` dialog, and click `Finish` to complete the registration (see Figure 3.14). Select `Cancel` to exit without completing the registration.

After registering your user identity and connection information, you are ready to log in to the Ark console (shown in Figure 3.15).

Copyright © 2011 - 2017 EnterpriseDB Corporation. All rights reserved.

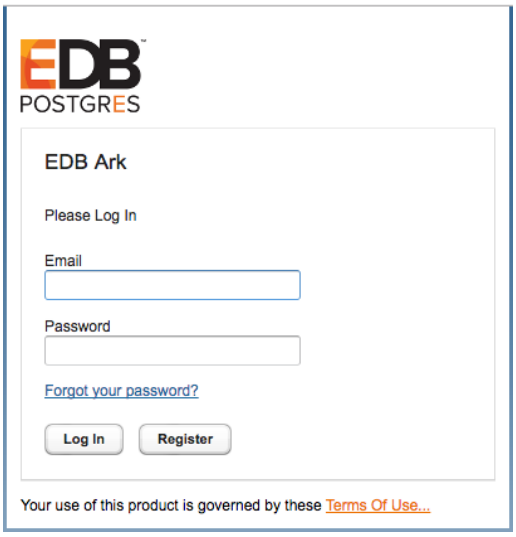


Figure 3.15 - The Login/Register dialog.

Provide the email address in the `Email` field, and the associated password in the `Password` field, and click `Log In` to connect to the Ark management console (shown in Figure 3.16).

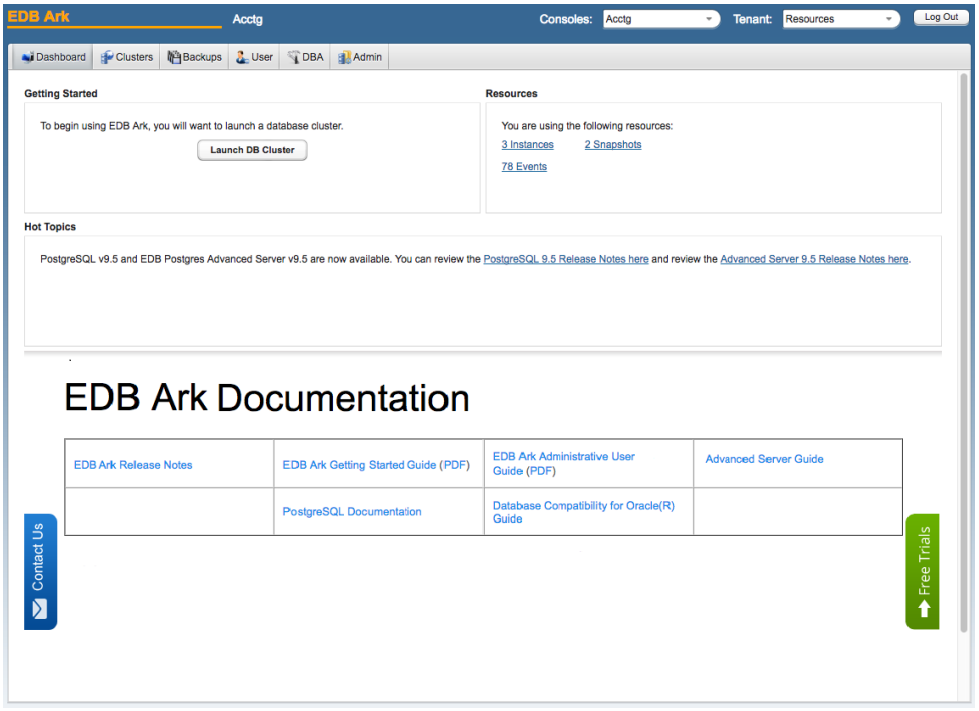


Figure 3.16 - The Dashboard tab of the Ark management console.

Copyright © 2011 - 2017 EnterpriseDB Corporation. All rights reserved.

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

1. Use the Ark console to define a server image for each server that will host a database cluster. For detailed information about using the Ark console to create server images, see Section 4.1.2.

2. Use the Ark console to create database engine definitions. For detailed information about defining a database engine, see Section 4.1.3.

Copyright © 2011 - 2017 EnterpriseDB Corporation. All rights reserved.

3.2 Installing EDB Ark for OpenStack

The installation instructions that follow describe the Ark console 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.

To install EDB Ark on an OpenStack host, you must:

1. Import the EDB Ark Image. For more information, see Section 3.2.1.
2. Create the EDB Ark Security Group. For more information, see Section 3.2.2.
3. Launch the Ark console instance. For more information, see Section 3.2.3.
4. Assign a floating IP address to the instance. For more information, see Section 3.2.4.
5. Modify the `ppcd.properties` file to configure the Ark console. For more information, see Section 3.2.5.
6. Deploy the Ark console. For more information, see Section 3.2.6.
7. Configure OpenStack user accounts. For more information, see Section 3.2.7.
8. Connect to the Ark console. For more information, see Section 3.2.8.

Copyright © 2011 - 2017 EnterpriseDB Corporation. All rights reserved.

3.2.1 Importing the EDB Ark Image on an OpenStack Host

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 3.17).

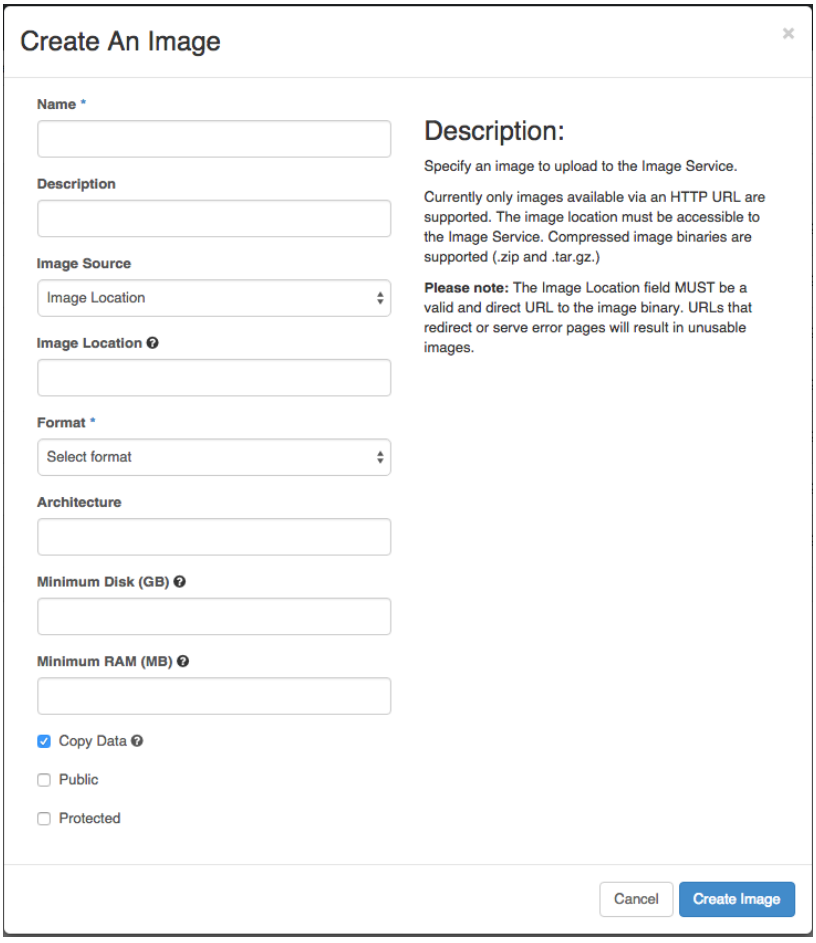


Figure 3.17 – The Create an Image dialog.

Copyright © 2011 - 2017 EnterpriseDB Corporation. All rights reserved.

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/v2.1
```

Copyright © 2011 - 2017 EnterpriseDB Corporation. All rights reserved.

```
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 2.1 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/

Copyright © 2011 - 2017 EnterpriseDB Corporation. All rights reserved.

3.2.2 Creating the EDB Ark Security Group

The security group for the Ark console must allow communication 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 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 3.18).

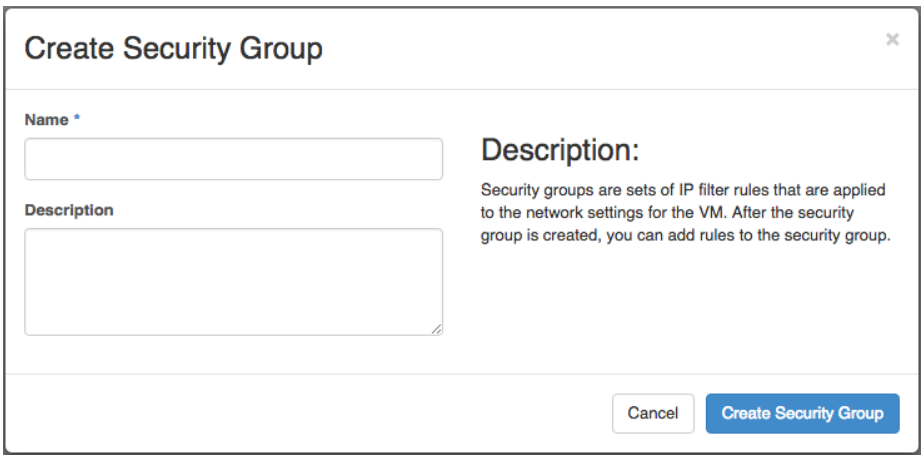


Figure 3.18 – The Create Security Group dialog.

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.

Copyright © 2011 - 2017 EnterpriseDB Corporation. All rights reserved.

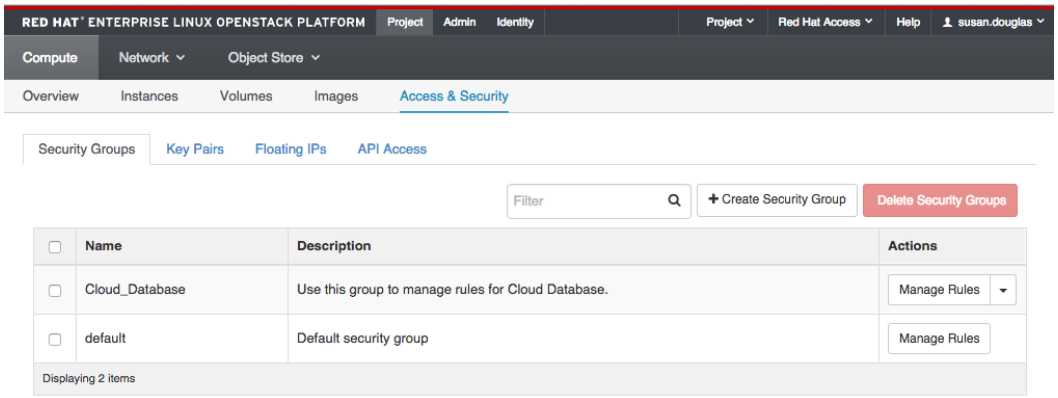


Figure 3.19 – The new group, displayed in the Security Groups list.

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 3.19). When the list of security group rules opens (see Figure 3.20), click the `+ Add Rule` button to access a dialog that allows you to add a new rule.

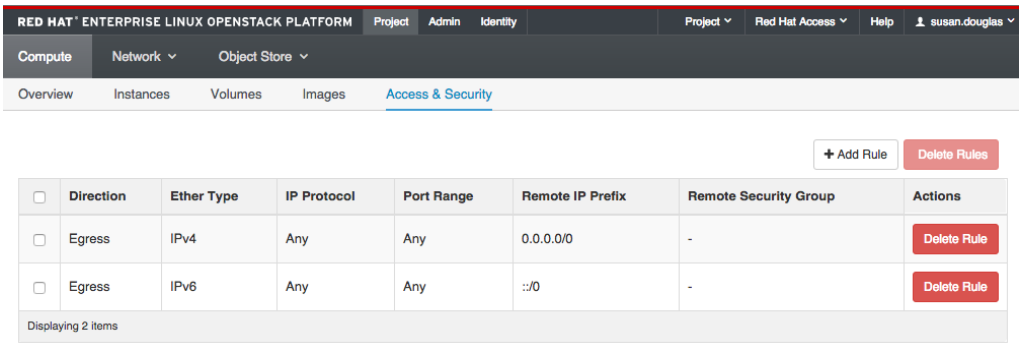


Figure 3.20 – The Security Group rules for the new security group.

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 7999	CIDR	0.0.0.0/0

Copyright © 2011 - 2017 EnterpriseDB Corporation. All rights reserved.

3.2.3 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 3.21).

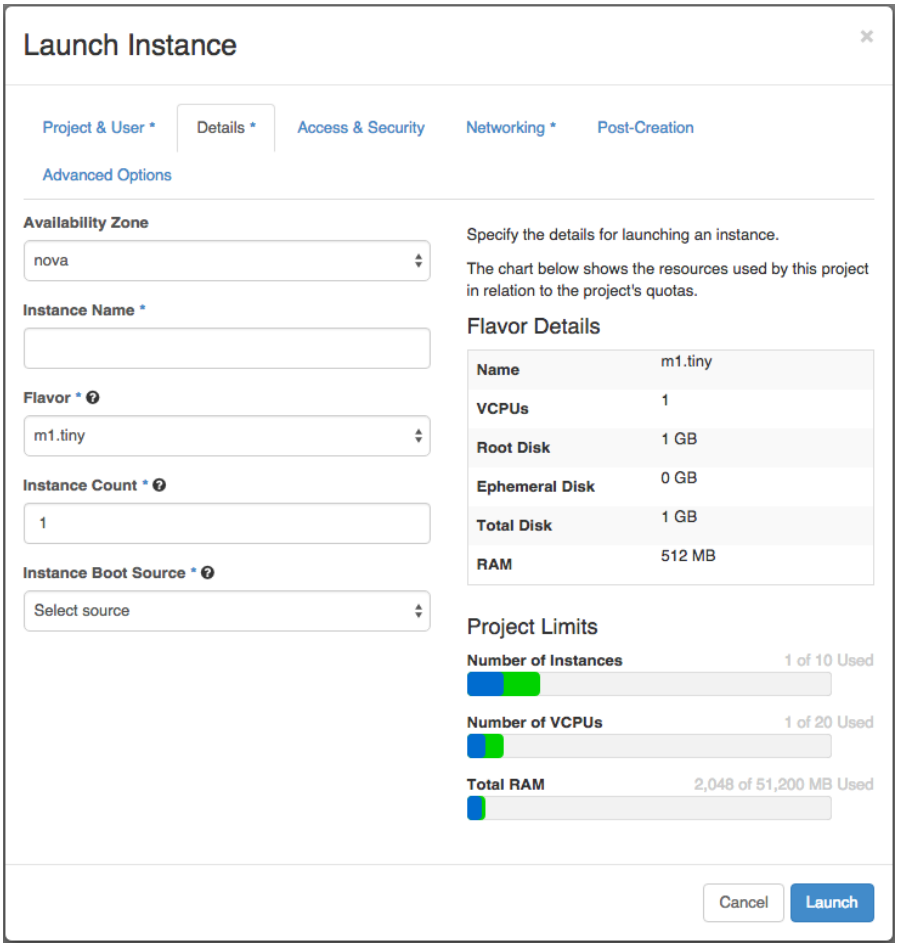


Figure 3.21 – *The Launch Instance dialog.*

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.

Copyright © 2011 - 2017 EnterpriseDB Corporation. All rights reserved.

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.

Copyright © 2011 - 2017 EnterpriseDB Corporation. All rights reserved.

3.2.4 Assign a Floating IP Address

When the instance launch completes, the new instance will be displayed on the `Instances` panel (as shown in Figure 3.22).

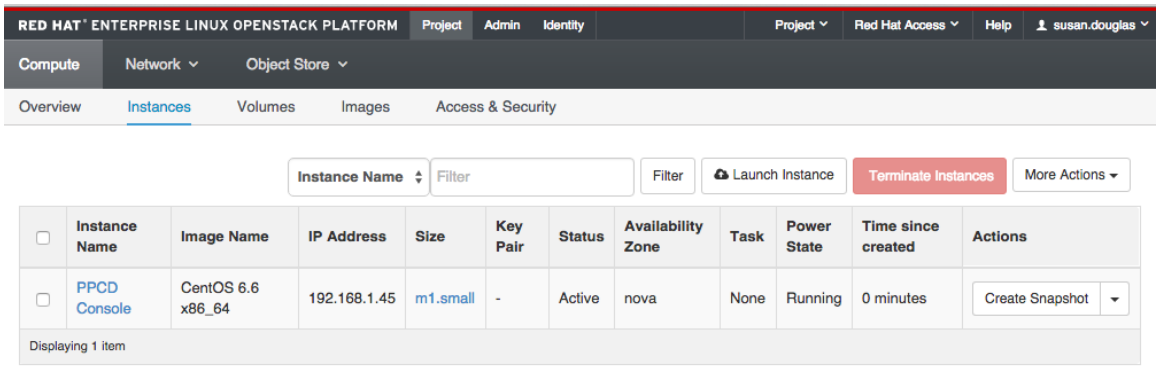


Figure 3.22*– The Instances dialog.*

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 3.23), use the `IP Address` drop-down listbox to select an IP address, or click the + button to allocate a new IP address.

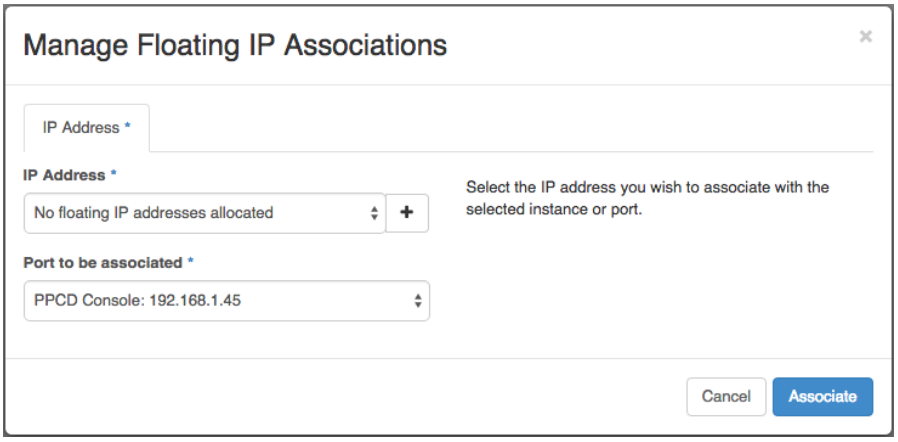


Figure 3.23*– The Manage Floating IP Associations dialog.*

Copyright © 2011 - 2017 EnterpriseDB Corporation. All rights reserved.

3.2.5 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.

Setting the Console Time Zone

After connecting with SSH, assume `root` privileges, and use the following commands to set the console time zone.

```
# rm /etc/localtime
# ln -s /usr/share/zoneinfo/time_zone /etc/localtime
# rm –f /etc/timezone
# ln –s /usr/share/zoneinfo/time_zone /etc/timezone
```

Copyright © 2011 - 2017 EnterpriseDB Corporation. All rights reserved.

Where *time_zone* specifies the time zone identifier that the console will use. To discover the available time zones for your system, you can use the command:

```
ls -l /usr/share/zoneinfo/
```

Then, restart the console's Postgres server, and then the GlassFish server:

```
# /etc/init.d/postgresql-9.6 restart
# su - ppcd
$ asadmin restart-domain
```

Then, use your choice of editor to modify the `ppcd.properties` file.

3.2.5.1 Configuring the ppcd.properties File on an OpenStack Host

You must supply configuration information in the `ppcd.properties` file before deploying the Ark console. Modify the `ppcd.properties` file (located in `/var/ppcd/`), 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.

Likewise, parameters that are prefaced with `aws` have values that correspond to values specified on the Amazon AWS management console. The value specified on the Amazon AWS Management console 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.

```
# 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= 0f42d1934a1a19f3d25d6288f2a3272c6143fc5d
# DB name to connect to
```

Copyright © 2011 - 2017 EnterpriseDB Corporation. All rights reserved.

```
# console.db.name=postgres
```

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

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 Ark user account (`ppcd`).

```
# 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 Ark user account (`ppcd`) 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
```

On an Amazon hosted console, you can use the `console.db.backup.container` and `console.db.backup.folder` parameters to specify the name of an Amazon S3 bucket to which console backups will be copied, and a console-specific folder name. If no value is specified for `console.db.backup.folder`, the value will default to `default`.

```
# Optional bucket name in which to store console backups
# console.db.backup.container=

# Unique name for the console backup folder that identifies this
# console, i.e. 'dev.console'. Default name is 'default'
# console.db.backup.folder=
```

Please note: Your AWS S3 backup container name must be unique when compared to the names of *all* other AWS containers. Including account specific information in the bucket identifier may help you create a unique name; for example:

`account-name.console.db.backup.container`

For more information about forming a bucket name, please consult the Amazon documentation at:

http://docs.aws.amazon.com/AmazonS3/latest/dev/BucketRestrictions.html

Copyright © 2011 - 2017 EnterpriseDB Corporation. All rights reserved.

Please note: backups are first created in the location specified in `console.db.backup.dir` before being copied to the container specified in `console.db.backup.container`. You must provide values for both parameters.

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
```

Please note: If you are using an AWS S3 bucket, your bucket name must be unique when compared to the names of *all* other AWS buckets. Including account specific information in the bucket identifier may help you create a unique name; for example:

Copyright © 2011 - 2017 EnterpriseDB Corporation. All rights reserved.

account-name.wal.archive.container

For more information about forming a bucket name, please consult the Amazon documentation at:

http://docs.aws.amazon.com/AmazonS3/latest/dev/BucketRestrictions.html

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
```

Copyright © 2011 - 2017 EnterpriseDB Corporation. All rights reserved.

```
service.account.password=password
```

Amazon AWS specific properties

The parameters listed in the `OpenStack specific properties` section will not apply to those consoles that are installed on an OpenStack host.

```
# the IAM role for the AWS service account
aws.service.account.rolearn=iam_role_arn
```

Use the `aws.service.account.externalid` parameter to specify the Amazon external ID that should be used by the Ark service user (`ppcd`).

```
# the external ID for the IAM role for the AWS service account
aws.service.account.externalid=iam_role_externalId
```

Use the `aws.region` parameter to specify the Amazon region in which Ark clusters will reside.

```
# the AWS region hosting your EDB Ark console (i.e. us-east-1)
aws.region=region_name
```

Use the `aws.cross.account.accesskey` parameter to specify the Amazon `AWS_ACCESS_KEY_ID` associated with the AWS role used for account administration.

```
# the AWS IAM cross account access key
aws.cross.account.accesskey=accesskeyid
```

Use the `aws.cross.account.secretkey` parameter to specify the Amazon `AWS_SECRET_ACCESS_KEY` associated with the AWS role used for account administration.

```
# the AWS IAM cross account secret key
aws.cross.account.secretkey=secretkeyid
```

If your console uses an Amazon AWS backing host, you can use the `self.registration.enabled` parameter to instruct the Ark console to enable or disable self-registration for Ark users.

If `self.registration.enabled` is set to `false`, an administrative user must register each Ark console user in the Ark administrative console.

If `self.registration.enabled` is set to `true`, the Ark console login dialog will display a `Register` button. An unregistered console user can use the `Register` button to access a dialog that allows them to register their own user account, and access the console. To successfully register, the user must know a valid Amazon Role ARN that will be associated with their identity.

Copyright © 2011 - 2017 EnterpriseDB Corporation. All rights reserved.

```
# Self registration enabled
self.registration.enabled=false
```

Display properties

Use the `console.dashboard.docs` and `console.dashboard.hot.topics` parameters to specify the source of the content that will be displayed on the `Dashboard` tab of the Ark console:

- If your cluster resides on a network with Internet access, set the parameters to `DEFAULT` to display content (alerts and documentation) from EnterpriseDB.
- If you would like the `Dashboard` tab to display alternate content, use the parameters to provide the URL of the content.
- If your cluster has limited access to the Internet, or if you wish to not display content on the `Dashboard` tab, leave the parameter values empty.

```
# these properties allow you to control the dashboard content.
# Legal values:
#       DEFAULT = load the default pages from enterprisedb.com
#       <unset> = don't load anything
#       <url>   = load alternate content at specified url
console.dashboard.docs=DEFAULT
console.dashboard.hot.topics=DEFAULT
```

3.2.6 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; when connecting to the console, use your OpenStack user name and password and an SSL encrypted browser connection

Copyright © 2011 - 2017 EnterpriseDB Corporation. All rights reserved.

(`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:

3. 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 4.1.2.

4. Use the Ark console to create database engine definitions. For detailed information about defining a database engine, see Section 4.1.3.

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 3.2.7.

Copyright © 2011 - 2017 EnterpriseDB Corporation. All rights reserved.

3.2.7 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 grant 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.

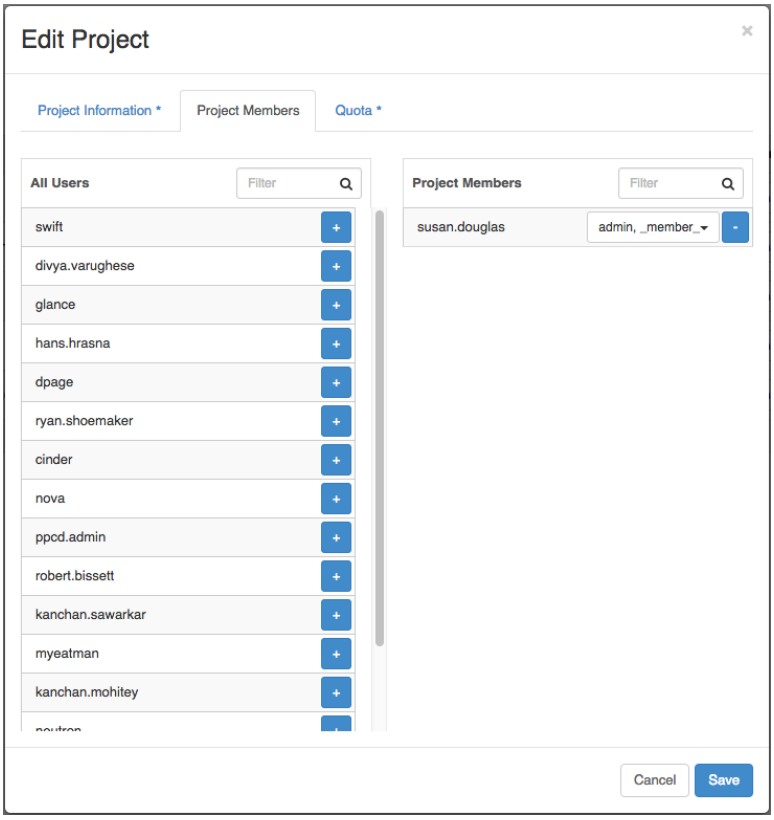


Figure 3.24 – The Edit Project dialog.

The `Edit Project` dialog opens, displaying the `Project Members` tab (see Figure 3.24).

- 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.

Copyright © 2011 - 2017 EnterpriseDB Corporation. All rights reserved.

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 3.25).

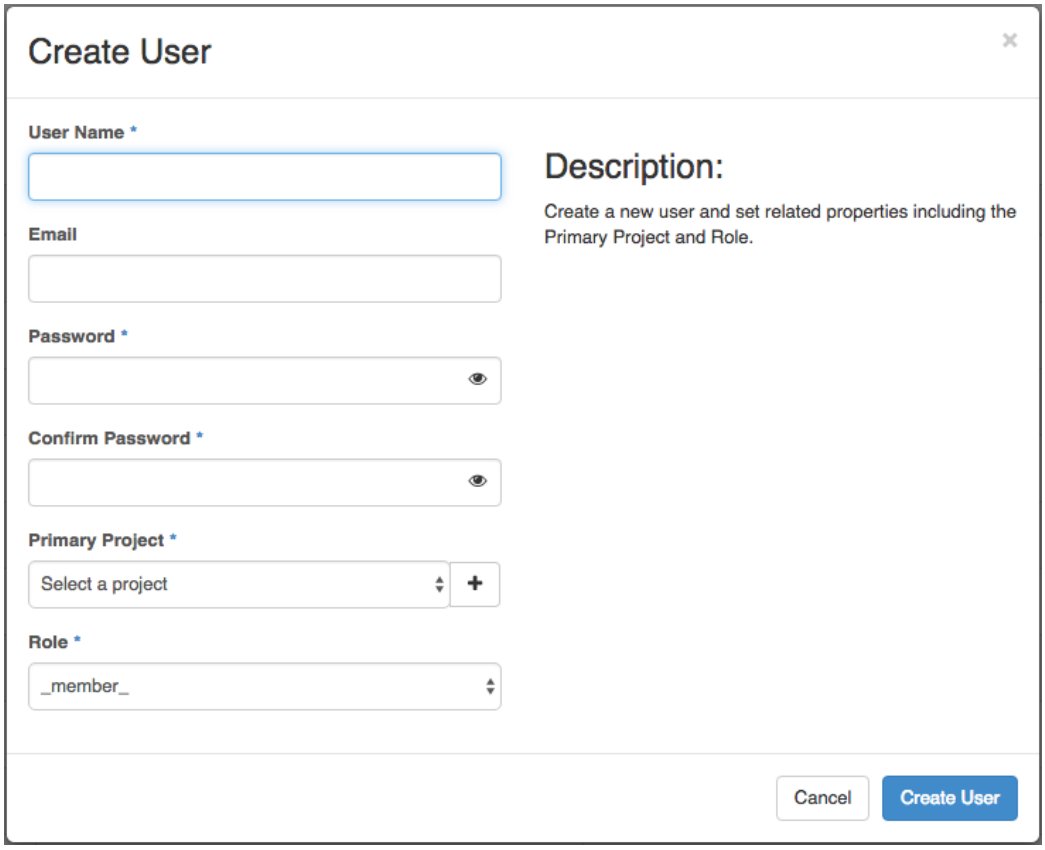


Figure 3.25 – The Create User dialog.

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.

Copyright © 2011 - 2017 EnterpriseDB Corporation. All rights reserved.

- 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

Copyright © 2011 - 2017 EnterpriseDB Corporation. All rights reserved.

3.2.8 Connecting to the Administrative Console on an OpenStack Host

When you navigate to the URL of the installed Ark console that uses OpenStack to host clusters, the console will display a login dialog (see Figure 3.26).

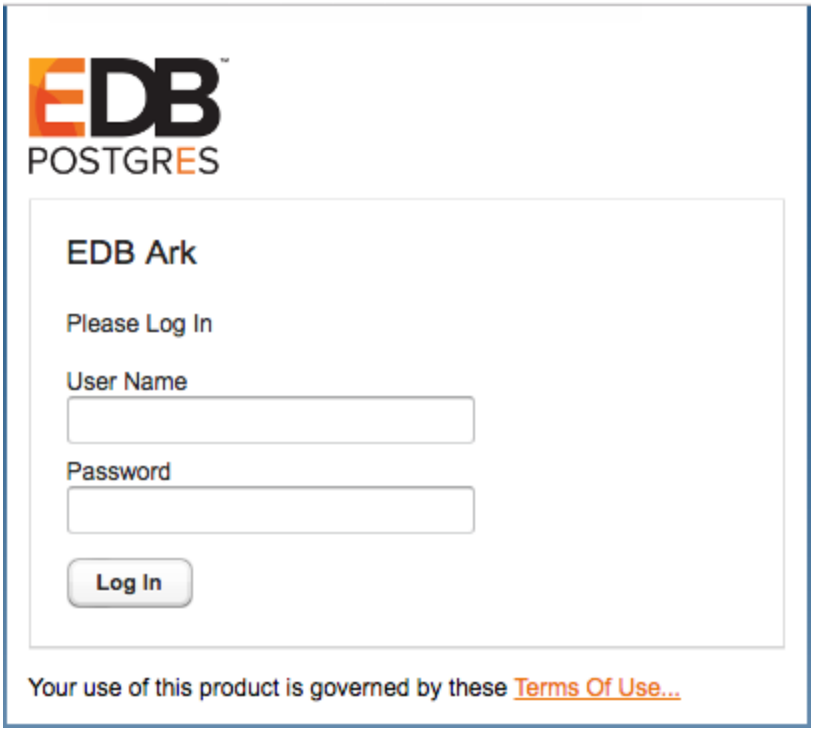


Figure 3.26 - The Login dialog.

Enter the name of an administrative 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 an OpenStack administrative role, the Ark console will include the `DBA` tab and the `Admin` tab (as shown in Figure 3.27).

Copyright © 2011 - 2017 EnterpriseDB Corporation. All rights reserved.

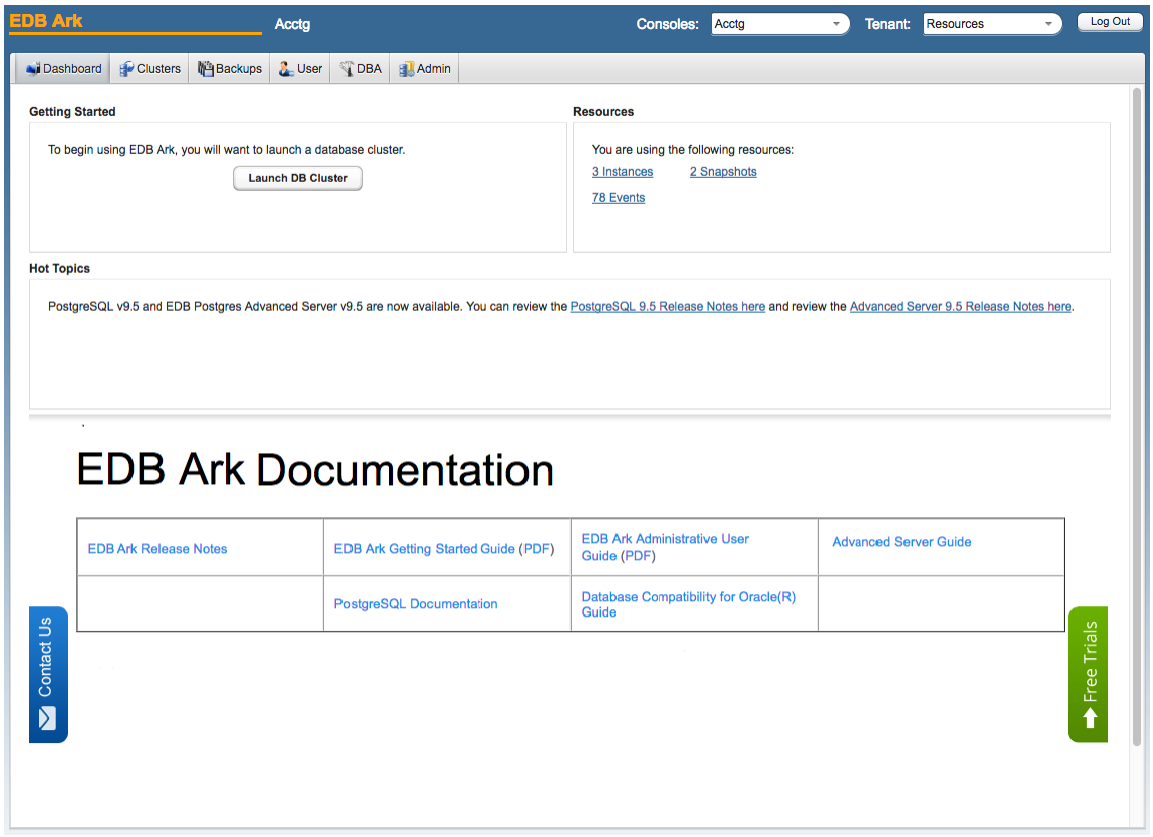


Figure 3.27 - The EDB Ark Administrator's console.

After connecting to the Ark console, you should:

- Update the `User` tab, providing a `Notification Email`. 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.

Copyright © 2011 - 2017 EnterpriseDB Corporation. All rights reserved.

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 Ark console, the console will display a login dialog (see Figure 4.1).

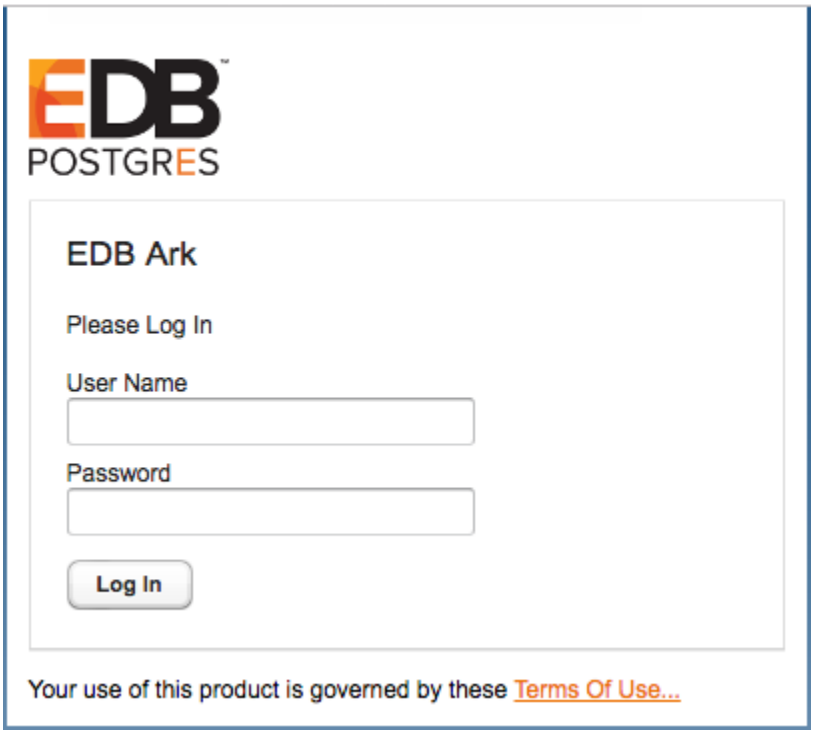


Figure 4.1 - The Login dialog.

Enter the name of an administrative user in the `User Name` field, and the associated password in the `Password` field, and click `Login` to connect to the Ark console. The console opens as shown in Figure 4.2.

Copyright © 2011 - 2017 EnterpriseDB Corporation. All rights reserved.

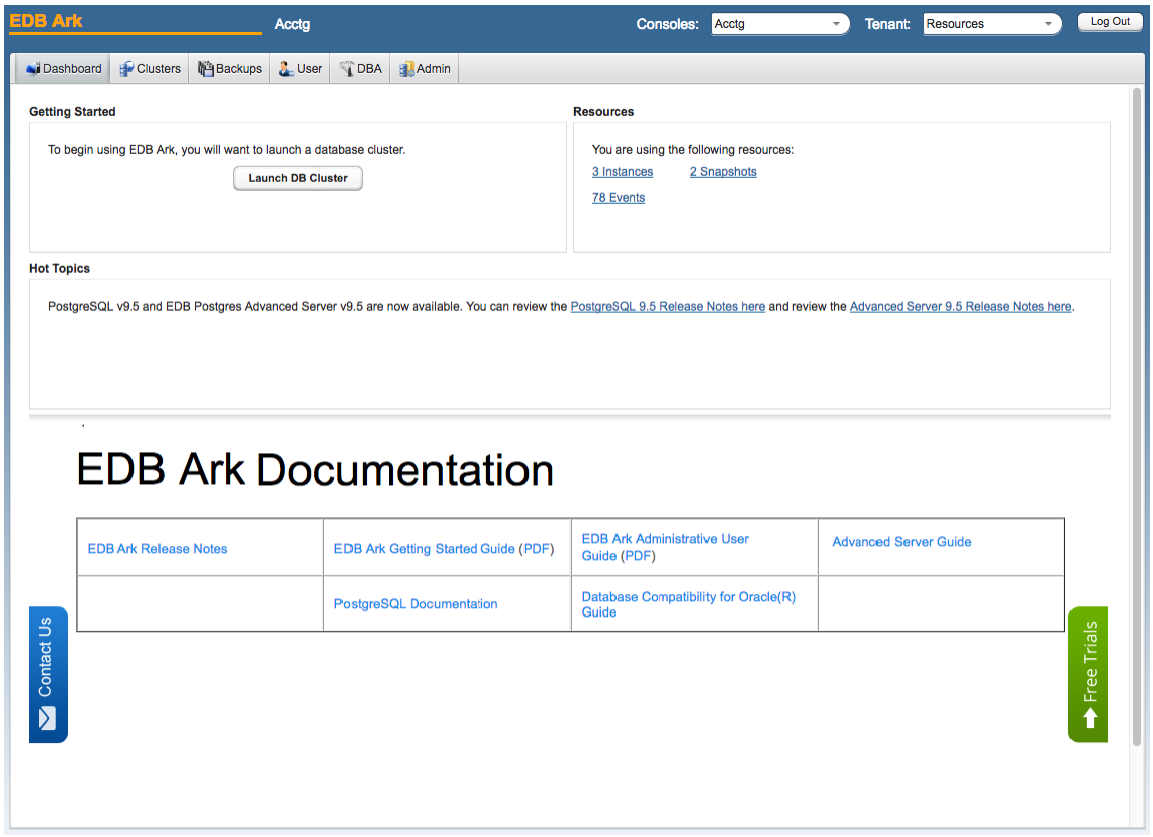


Figure 4.2 - The EDB Ark Administrator's console.

Copyright © 2011 - 2017 EnterpriseDB Corporation. All rights reserved.

4.1 Using the Admin Tab

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

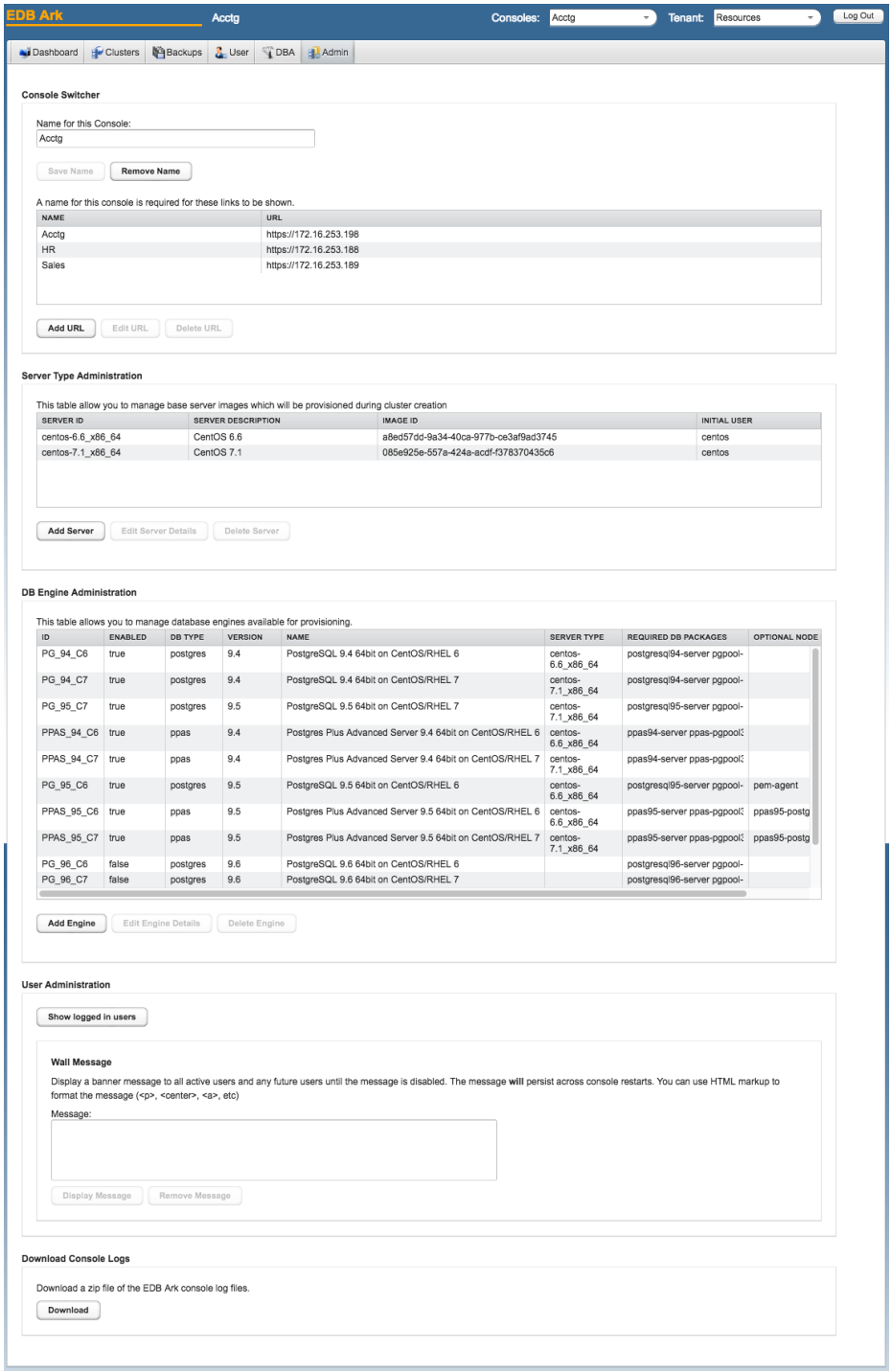


Figure 4.3 – The Admin tab

Copyright © 2011 - 2017 EnterpriseDB Corporation. All rights reserved.

Console Switcher

Use the fields in the `Console Switcher` box to:

- Make a console available through the `Consoles` drop-down listbox on the Ark console.

For information about using the Console Switcher features, see Section 4.1.1.

Server Type Administration

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)

For information about adding additional servers, see Section 4.1.2.

DB Engine Administration

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 server images that correspond to a server that resides on your system. For more information about defining a server type, see Section 4.1.2.

- Use the `Edit Engine Details` button to modify existing engine definitions to specify a 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 4.1.3.

User Administration

Options in the `User Administration` box allow you to:

- Add, modify, or delete user accounts (for an AWS instance).
- Access a list of currently connected users.
- Display a banner message to connected users.

For information about user administration options, see Section 4.1.4.

Copyright © 2011 - 2017 EnterpriseDB Corporation. All rights reserved.

Download Console Logs

Use the `Download` button in the `Download Console Logs` box to download a zip file that contains the server logs for the underlying application server. You can use the log file to confirm changes to server status or verify server activity.

For more information, see Section 4.1.5.

Copyright © 2011 - 2017 EnterpriseDB Corporation. All rights reserved.

4.1.1 Using the Console Switcher Feature

The console switcher provides convenient access to a list of user-defined console names and their associated addresses. When you select a name from the `Consoles` drop-down listbox (see Figure 4.4), the Ark console opens a browser tab and navigates to the address associated with the name.

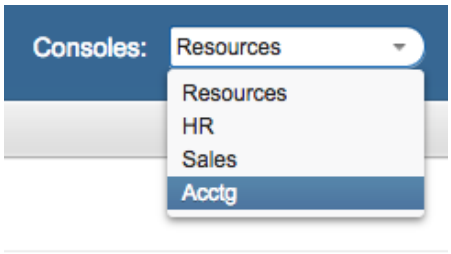


Figure 4.4 – The Consoles drop-down.

Use the `Console Switcher` section of the `Admin` tab to manage the console names and addresses that are displayed in the `Consoles` drop-down (see Figure 4.5).

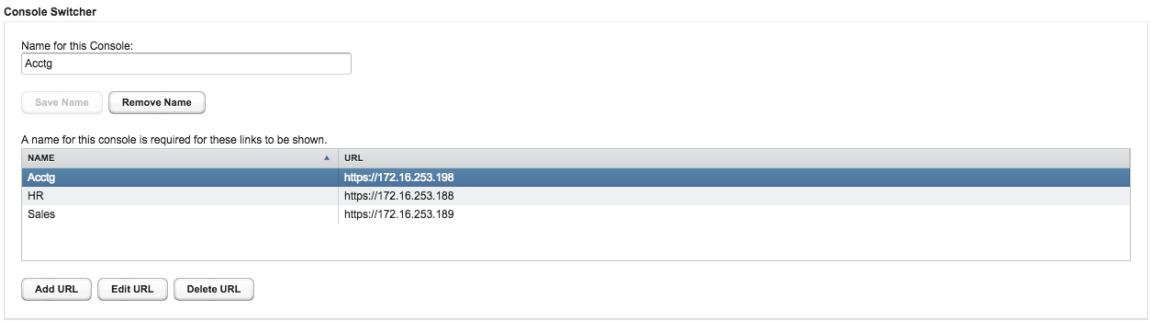


Figure 4.5 – The Console Switcher section of the Admin tab.

To enable the `Consoles` drop-down, you must first provide a name for the console to which you are currently connected in the `Name for this Console` field on the Admin tab (see Figure 4.6).

Figure 4.6 – The Consoles drop-down.

Copyright © 2011 - 2017 EnterpriseDB Corporation. All rights reserved.

After providing the console name, click the `Save Name` button to display the name of the console in the upper-left corner of the Ark console, and in the `Consoles` drop-down. To add a shortcut to another console, click the `Add URL` button; the `Add URL` dialog opens as shown in Figure 4.7.

Figure 4.7 – The Add URL dialog.

Use the `Add URL` dialog to provide information about the console for which you are creating a `Consoles` entry:

- Provide a user-friendly name in the `Name` field.
- Provide the URL of the console in the `Url` field; please note that the URL must be prefixed with the http protocol identifier.

When you're finished, click the `Save` button to add the console to the list displayed on the `Consoles` drop-down.

To modify an entry in the `Consoles` drop-down, highlight the name of the console in the `NAME` column and click the `Edit URL` button. The `Edit URL` dialog opens (as shown in Figure 4.8).

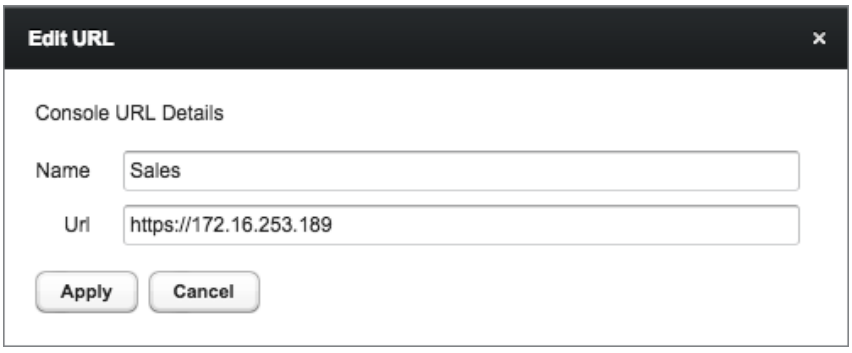


Figure 4.8 – The Edit URL dialog.

Copyright © 2011 - 2017 EnterpriseDB Corporation. All rights reserved.

After modifying the console details on the `Edit URL` dialog, click the `Apply` button to preserve the changes. Click `Cancel` to exit the dialog without saving your changes.

To remove a URL, highlight the name of the URL in the `NAME` column and click the `Delete URL` button. A dialog will open, asking you to confirm that you wish to delete the URL (see Figure 4.9).

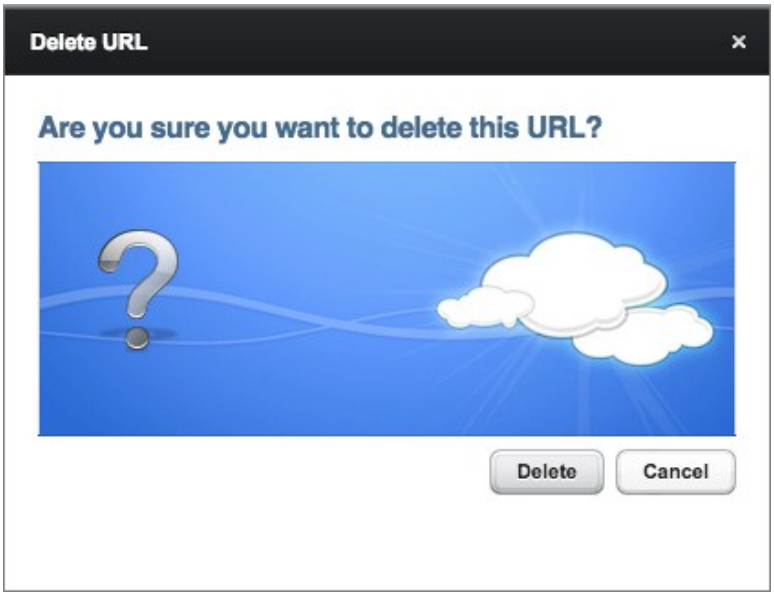


Figure 4.9 – The Edit URL dialog.

Click the `Delete` button to confirm that you want to remove the entry from the `Consoles` drop-down and delete the entry from the `Console Switcher` table, or click `Cancel` to exit the dialog without deleting the entry.

Copyright © 2011 - 2017 EnterpriseDB Corporation. All rights reserved.

4.1.2 Managing Server Images

A server definition describes the virtual machine that will host an instance of Advanced Server or PostgreSQL. Use the `Server Type Administration` section of the `Admin` tab to manage server images (see Figure 4.10).

Figure 4.10 – The Server Type Administration section of the Admin tab.

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 4.11 opens.

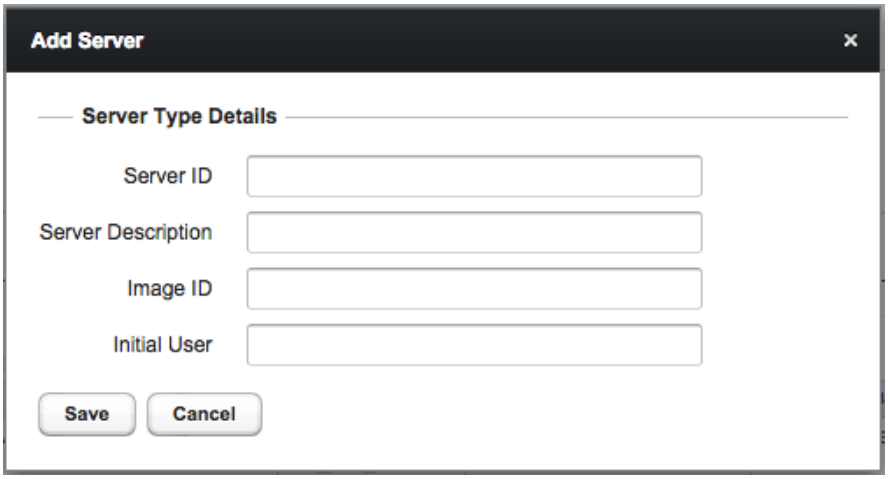


Figure 4.11 – The Add Server dialog.

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.

Copyright © 2011 - 2017 EnterpriseDB Corporation. All rights reserved.

- Use the Server Description field to provide a description of the server image.

- Use the Image ID field to provide the Image ID of the server image.

  On OpenStack, 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. The image must be either a public image, or available to all tenants or roles that are allowed to run EDB Ark clusters.

  If you are using Amazon, provide the AMI ID in the Image ID field. You can locate the AMI ID on the AWS management console AMIs dashboard. Please note: you should use a server from a trusted source, with a virtualization type of hvm.

- 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 4.12) and modify the properties of a server.

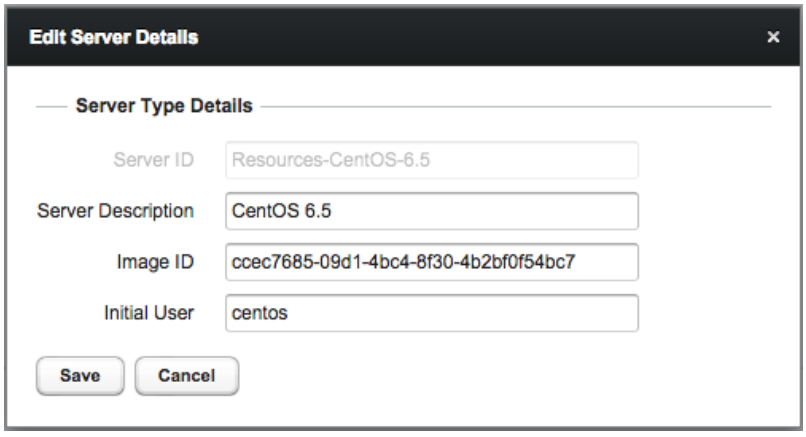


Figure 4.12 – The Edit Server dialog.

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

Copyright © 2011 - 2017 EnterpriseDB Corporation. All rights reserved.

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 4.13).

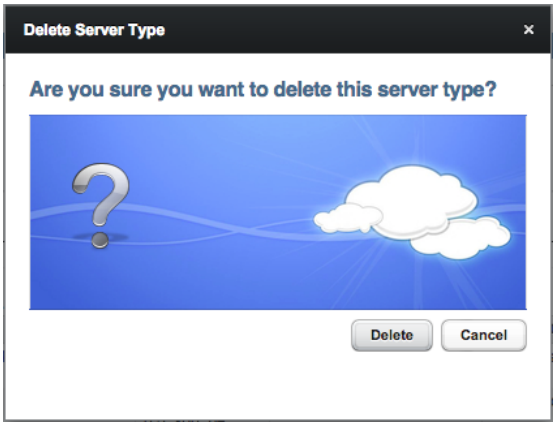


Figure 4.13 – The Delete Server Type dialog.

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

Figure 4.14 – 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 4.14); before removing the server, you must delete any dependent engines.

Copyright © 2011 - 2017 EnterpriseDB Corporation. All rights reserved.

4.1.3 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. You can use the `DB Engine Administration` section of the `Admin` tab to create and manage database engines (see Figure 4.15).

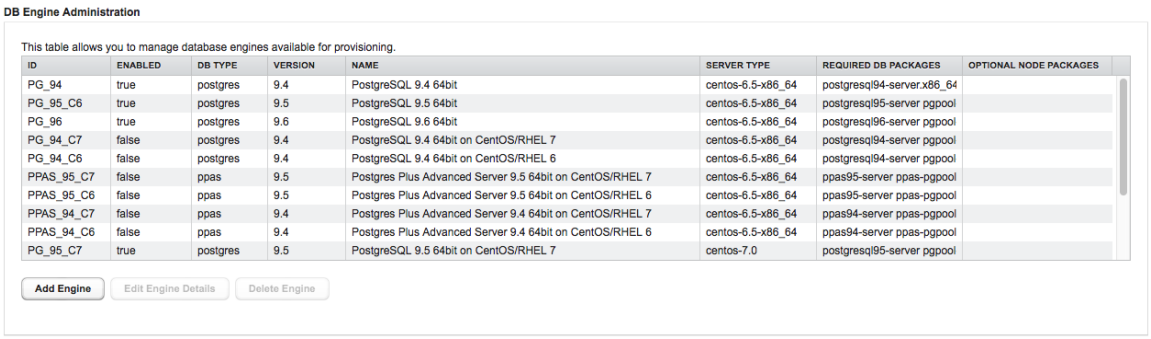


Figure 4.15 – The Server Type Administration section of the Admin tab.

Adding an Engine

Use the `Add Engine` dialog (see Figure 4.16) 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`.

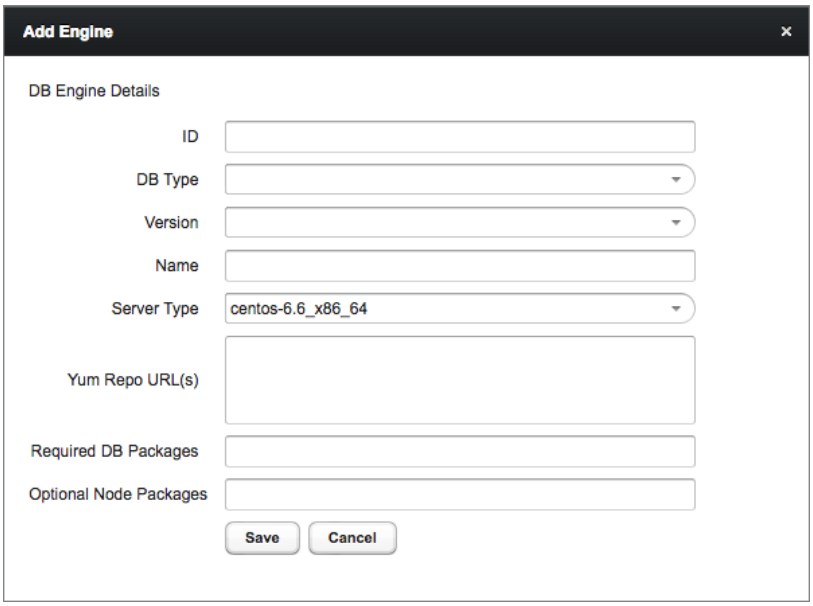


Figure 4.16 – The Add Engine dialog.

Copyright © 2011 - 2017 EnterpriseDB Corporation. All rights reserved.

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
  ```

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

  password specifies the password associated with the repository user. Please note that if your password contains special characters (such as a $), you may need to percent-encode the characters.

  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
  ```

  Where *x* specifies the specific Advanced Server version (e.g., 9.6).

  Advanced Server 9.6 dependencies are satisfied by RPM files from:

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

Copyright © 2011 - 2017 EnterpriseDB Corporation. All rights reserved.

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
```

Where *x* specifies the specific PostgreSQL version (e.g., 96 or 9.6).

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-pgpool34
  ppas95-pgpool34-extensions
  ```

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

  ```
  ppas95-server
  ppas-pgpool34
  ppas95-pgpool34-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:

Copyright © 2011 - 2017 EnterpriseDB Corporation. All rights reserved.

```
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 4.17).

Copyright © 2011 - 2017 EnterpriseDB Corporation. All rights reserved.

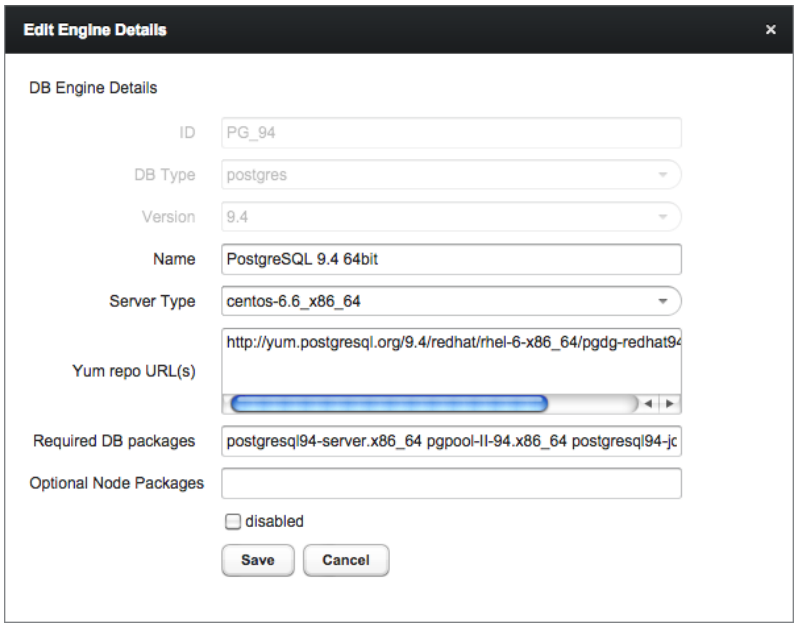


Figure 4.17 – The Edit Engine Details dialog.

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.

Please note that disabling an engine has no impact on any running clusters; it simple prevents users from creating new clusters with the engine. You can use this feature to phase out the use of older engines.

Deleting an Engine

Copyright © 2011 - 2017 EnterpriseDB Corporation. All rights reserved.

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 4.18).

Figure 4.18 – The Delete DB Engine dialog.

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

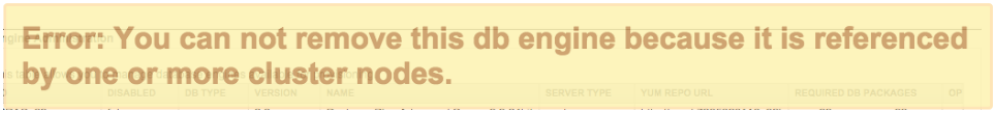


Figure 4.19 – The Delete DB Engine dialog.

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 4.19.

Copyright © 2011 - 2017 EnterpriseDB Corporation. All rights reserved.

4.1.3.1 Adding Packages to a Database Engine Definition

When you create a cluster, you select the engine that EDB Ark will use when provisioning the cluster. If you modify the engine description, adding the list of RPM packages that will be installed when that engine is provisioned, each node of any cluster provisioned with that engine will include the functionality of the supporting component.

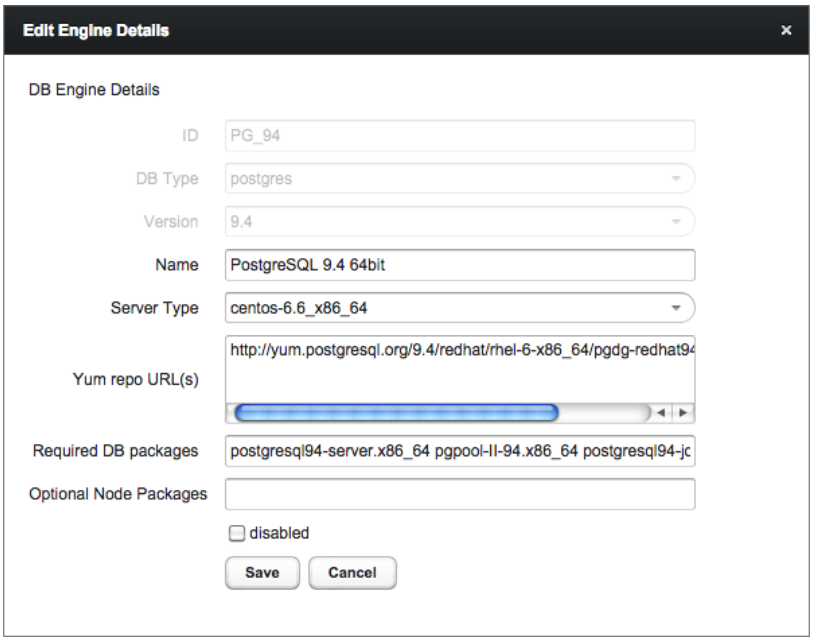


Figure 4.20 – The Edit Engine Details dialog.

4.1.3.1.1 Adding PostGIS to a Database Engine

To simplify PostGIS installation, add a list of the required RPM packages to the `Optional Node Packages` field of the `Edit Engine Details` dialog (see Figure 4.20). After installing PostGIS, you must create the PostGIS extensions. To provision replicas that contain the PostGIS functions, perform the installation and create the extensions on the master node of the cluster before adding replica nodes to your cluster.

To modify an engine description, use Administrator credentials to connect to the Ark console, and navigate to the `Admin` tab. Select an engine `ID` from the list of engines in the `DB Engine Administration` list, and click `Edit Engine Details`.

Copyright © 2011 - 2017 EnterpriseDB Corporation. All rights reserved.

Figure 4.21 – Modifying the Engine Details dialog.

When the `Edit Engine Details` dialog opens (see Figure 4.21), use the fields on the dialog to specify the repository information and the names of optional RPM packages that the installer should provision on each node of the cluster.

- The PostGIS RPM packages are distributed from the `enterprisedb tools` repository; by default, the `enterprisedb tools` repository is included in the `Yum Repo URL` field.

- Add the names of the PostGIS RPM packages to the `Optional Node Packages` field on the `Edit Engine Details` dialog.

  The PostGIS installation packages for Advanced Server 9.4 are:

  ```
  ppas94-postgis
  ppas94-postgis-core
  ppas94-postgis-docs
  ppas94-postgis-utils
  ```

  The PostGIS installation packages for Advanced Server 9.5 are:

  ```
  ppas95-postgis
  ppas95-postgis-core
  ppas95-postgis-docs
  ppas95-postgis-utils
  ```

Copyright © 2011 - 2017 EnterpriseDB Corporation. All rights reserved.

Any EDB Ark clusters that are subsequently provisioned with that engine will automatically include an installation of the PostGIS on all nodes of the cluster (see Figure 4.22).

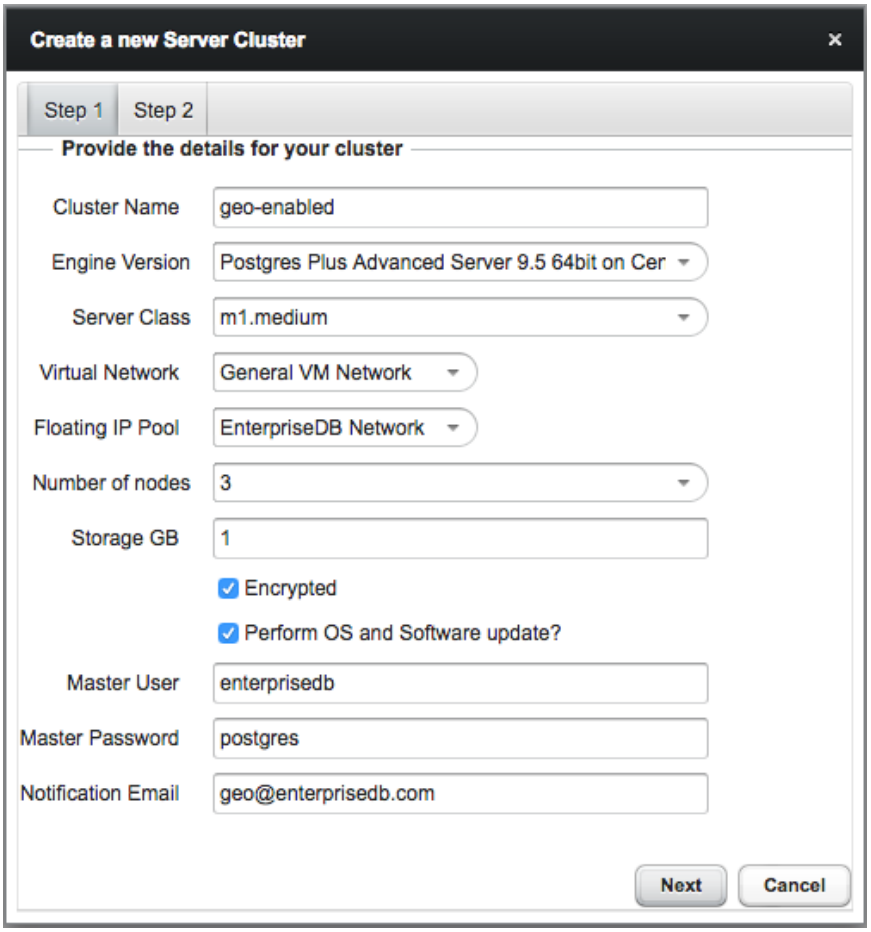


Figure 4.22 – Use the modified engine when provisioning a cluster.

For detailed information about creating a new server cluster, please see the *EDB Ark Getting Started Guide*, available through the EDB Ark `Dashboard` tab.

Copyright © 2011 - 2017 EnterpriseDB Corporation. All rights reserved.

Creating the PostGIS Extensions

You can use the psql client or the EDB Postgres Enterprise Manager (PEM) client to install the extensions. Before connecting with a client, an Administrator must open the listener port (by default, `5444`) of the node for connections.

The example that follows demonstrates creating the extensions with the PEM client on an Advanced Server host. The PEM client is installed with the Advanced Server graphical installer, and is also available for PostgreSQL users. For more information about the PEM client, visit:

https://www.enterprisedb.com/products/edb-postgres-platform/edb-postgres-enterprise-managerpem

The PEM client should be installed on and invoked from a local workstation.

To open the PEM client, navigate through the `Start` menu (on Linux) or the `Apps` menu (on Windows), selecting `Postgres Enterprise Manager v6` from the `Postgres Plus Advanced Server 9.5` menu (see Figure 4.23).

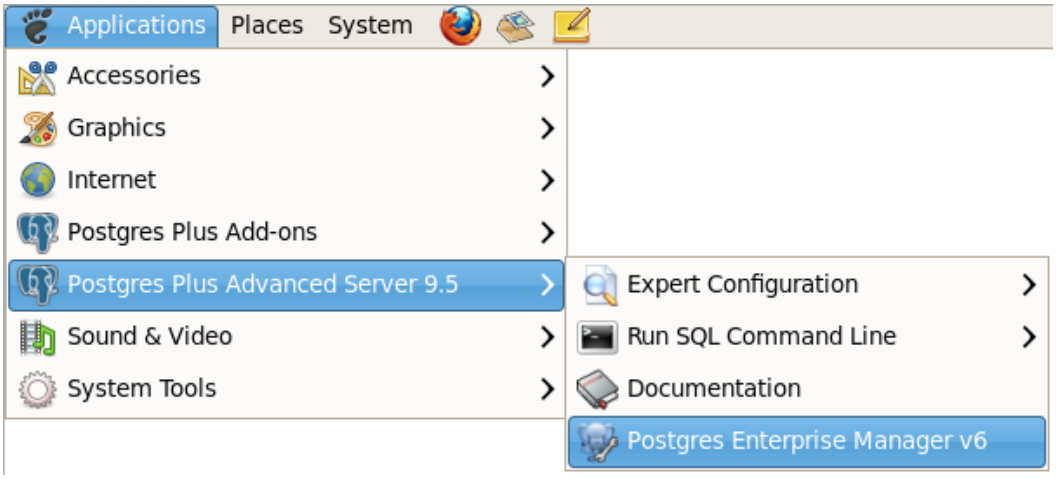


Figure 4.23 – Opening the PEM Client.

Before you can access an EDB Ark cluster, you must register the server; use the `New Server Registration` dialog to register the server. To open the `New Server Registration` dialog, select `Add Server` from the `File` menu.

Copyright © 2011 - 2017 EnterpriseDB Corporation. All rights reserved.

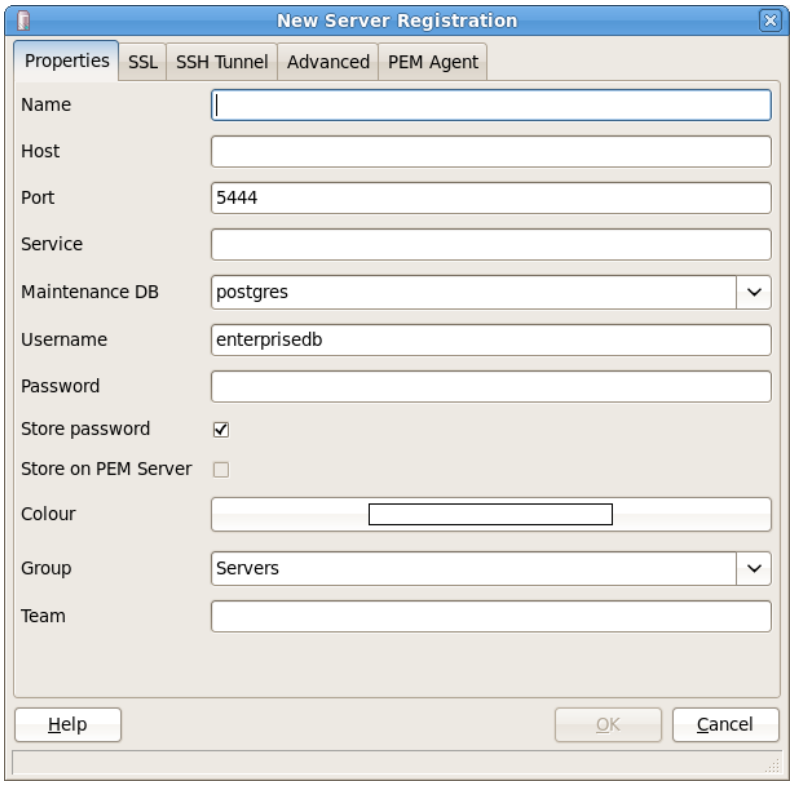


Figure 4.24 - Connecting to the EDB Ark host.

Provide information about the server in the `New Server Registration` dialog (see Figure 4.24):

- Specify a descriptive name of the EDB Ark cluster in the `Name` field.
- Provide the IP address of the server in the `Host` field. You can find the IP address in the `DNSNAME` column on the `Details` panel for the cluster on the EDB Ark console.
- Specify the `Port` through which you wish to connect to the server.

  If you are modifying a database or invoking administrative functions, you should connect to the master node's listener port, identified in the `DBPORT` column, on the `Details` panel of the `Clusters` tab. Before connecting to the server's listener port, an OpenStack administrator must modify the cluster's security group to allow connections from the connecting client.
- Select a maintenance database using the drop-down listbox in the `Maintenance DB` field. Select `edb` if you are connecting to an Advanced Server database, or `postgres` if you are connecting to a PostgreSQL database.
- Specify the role name that the PEM client should use when connecting in the `Username` field.
- Provide the password associated with that role, in the `Password` field.

Copyright © 2011 - 2017 EnterpriseDB Corporation. All rights reserved.

Click OK to connect to EDB Ark; once connected, the server will appear in the tree control in the PEM Object browser (shown in Figure 4.25).

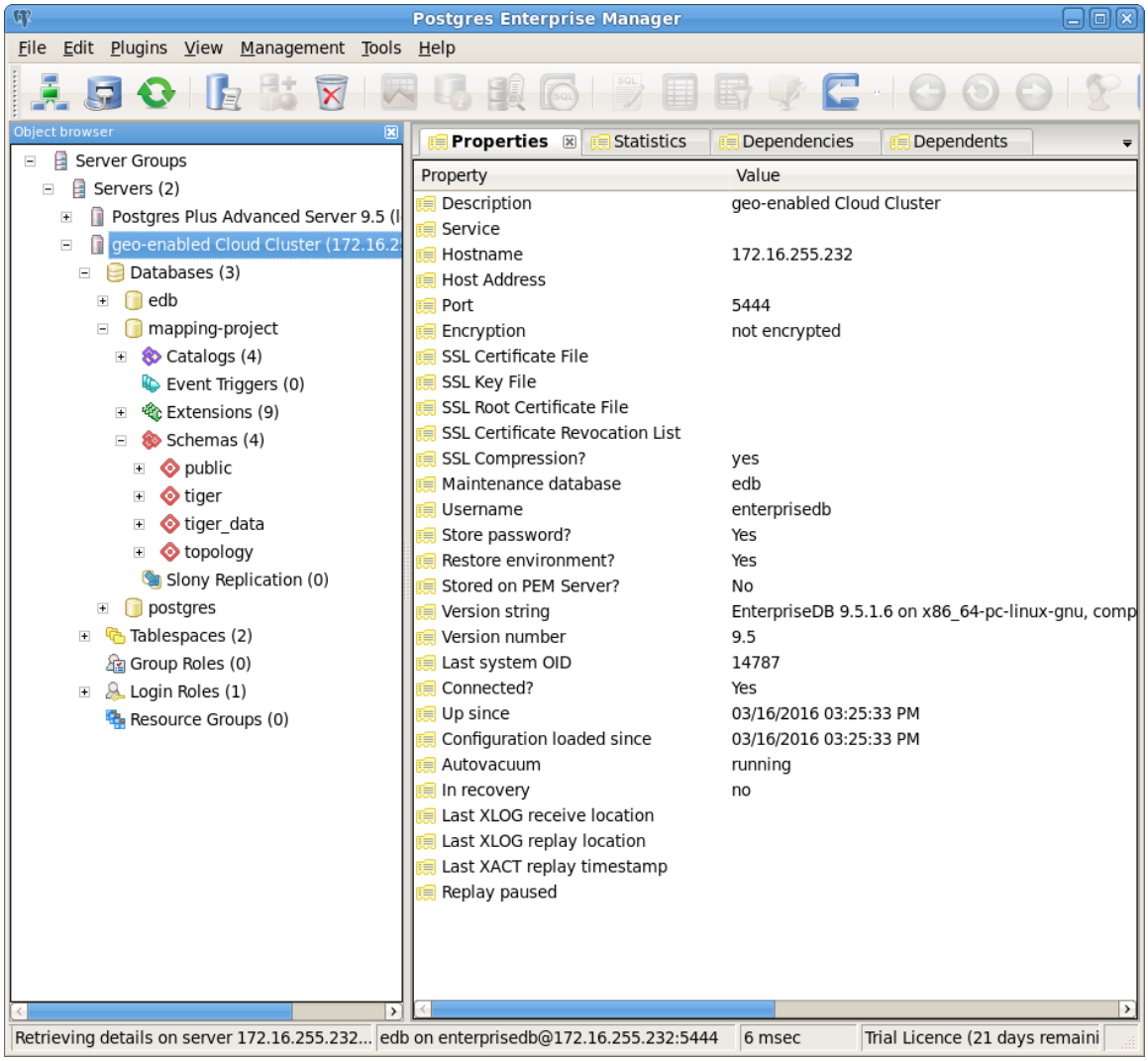


Figure 4.25 - The PEM client window, showing local and EDB Ark servers.

After connecting to the server with the PEM client, you should create a database in which to install the PostGIS extensions. The PostGIS extensions must be installed in each database in which you wish to use PostGIS functions.

Copyright © 2011 - 2017 EnterpriseDB Corporation. All rights reserved.

To use the PEM client to create a database, right click on the Databases node of the tree control (under the EDB Ark server), and select New Database... from the context menu. The New Database... dialog opens (as shown in Figure 4.26).

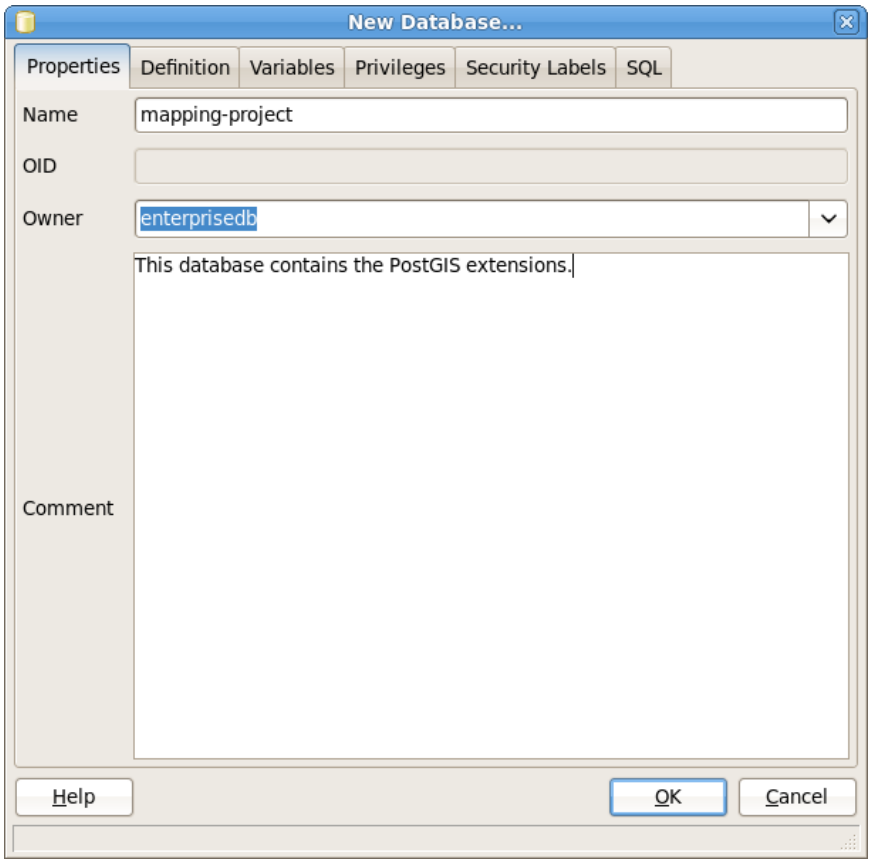


Figure 4.26 - The PEM client window, showing local and EDB Ark servers.

Provide details about the new database in the New Database... dialog. The New Database dialog is a point-and-click interface that allows you to implement options of the CREATE DATABASE command. For more information about the CREATE DATABASE command, please see:

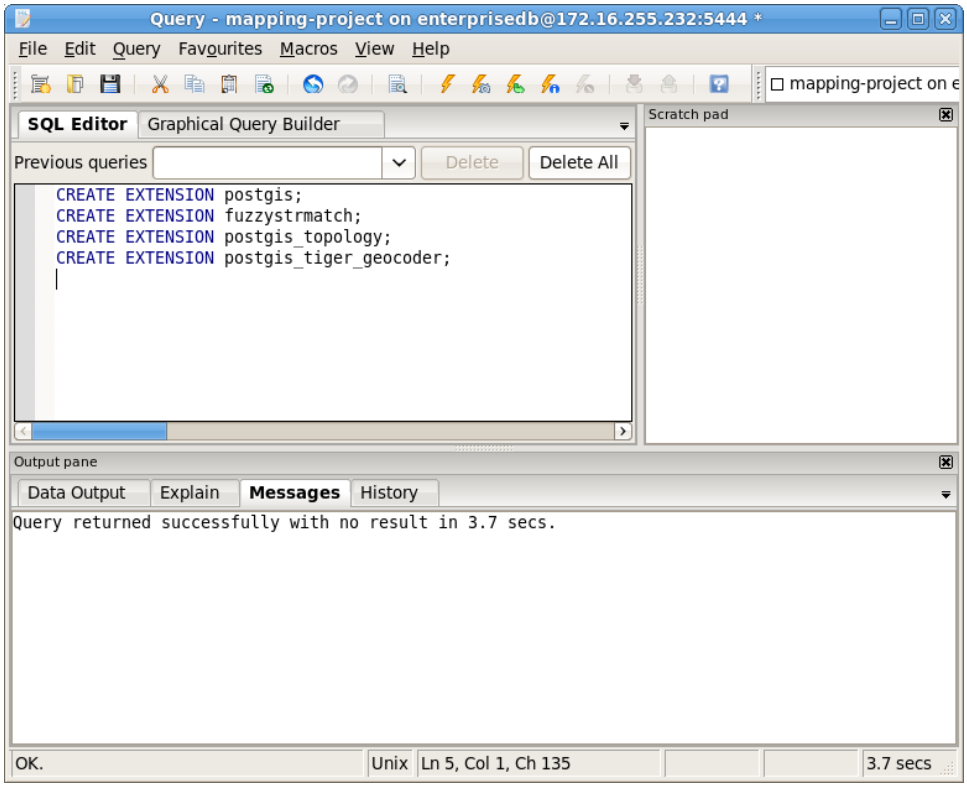
http://www.postgresql.org/docs/9.6/static/sql-createdatabase.html

Copyright © 2011 - 2017 EnterpriseDB Corporation. All rights reserved.

To use the PEM Query Tool to install the extensions, highlight the name of the database into which you are installing the extensions in the Object browser tree control, and select Query tool from the Tools menu. When the Query Tool opens, enter the commands that create the extensions in the SQL Editor pane:

```
CREATE EXTENSION postgis;
CREATE EXTENSION fuzzystrmatch;
CREATE EXTENSION postgis_topology;
CREATE EXTENSION postgis_tiger_geocoder;
```

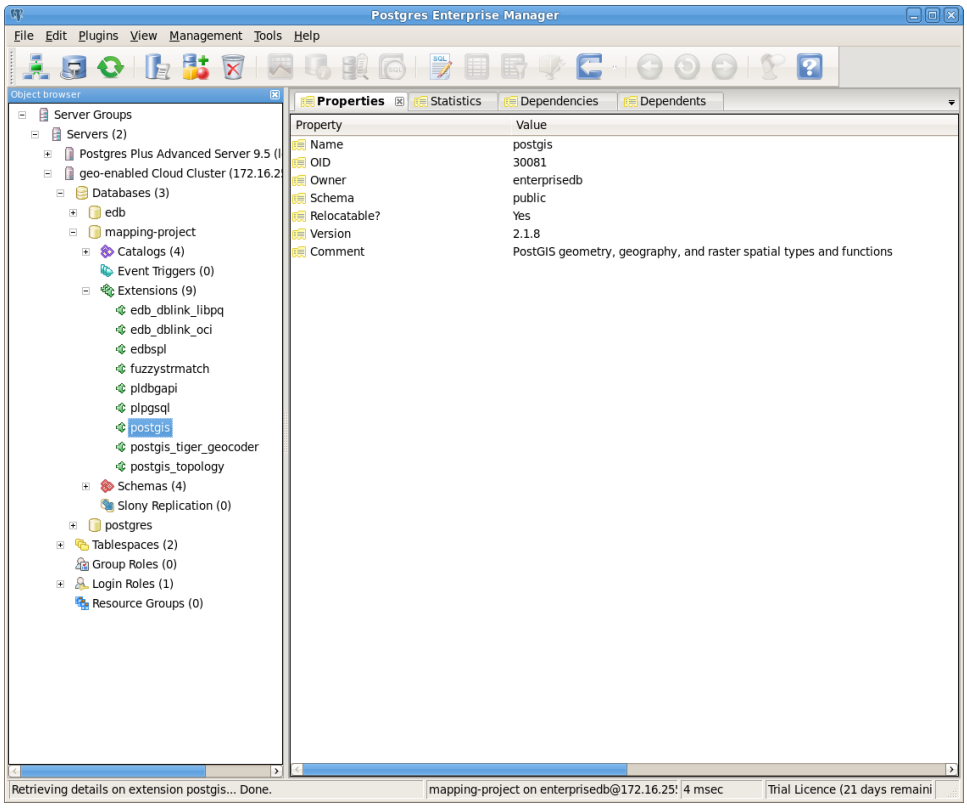
Then, execute the commands; to execute the commands, select Execute from the Query menu, or click the Execute icon.



4.27 - Creating the PostGIS extensions with the query tool.

The output pane will confirm that the extensions were created successfully (see Figure 4.27).

Copyright © 2011 - 2017 EnterpriseDB Corporation. All rights reserved.



4.28 – The PostGIS extensions are listed in the extensions node.

The PostGIS extensions will be displayed in the `Extensions` node of the `Object browser` tree control (see Figure 4.28).

Copyright © 2011 - 2017 EnterpriseDB Corporation. All rights reserved.

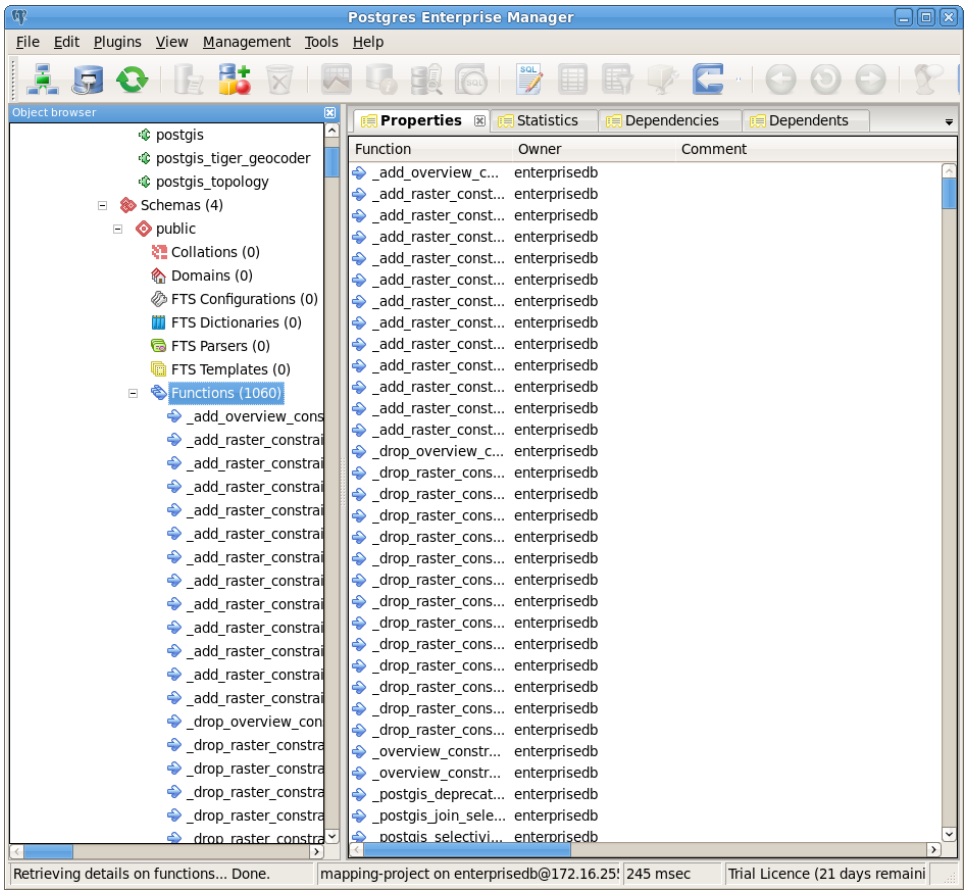


Figure 4.29 - The PostGIS functions are in the public schema under the database.

The PostGIS functions are installed in the `public` schema of the database (see Figure 4.29).

For detailed information about using PostGIS, please see the project documentation at:

http://postgis.net/documentation/

Copyright © 2011 - 2017 EnterpriseDB Corporation. All rights reserved.

4.1.3.1.2 Adding the PEM Agent to a Database Engine

The PEM agent is responsible for executing tasks and reporting statistics from a monitored Postgres instance to the PEM server. The PEM agent can be installed with an RPM package; if you have administrative access to the Ark console, you can configure the console to automatically install the PEM agent on Master and Standby nodes when provisioning a cluster.

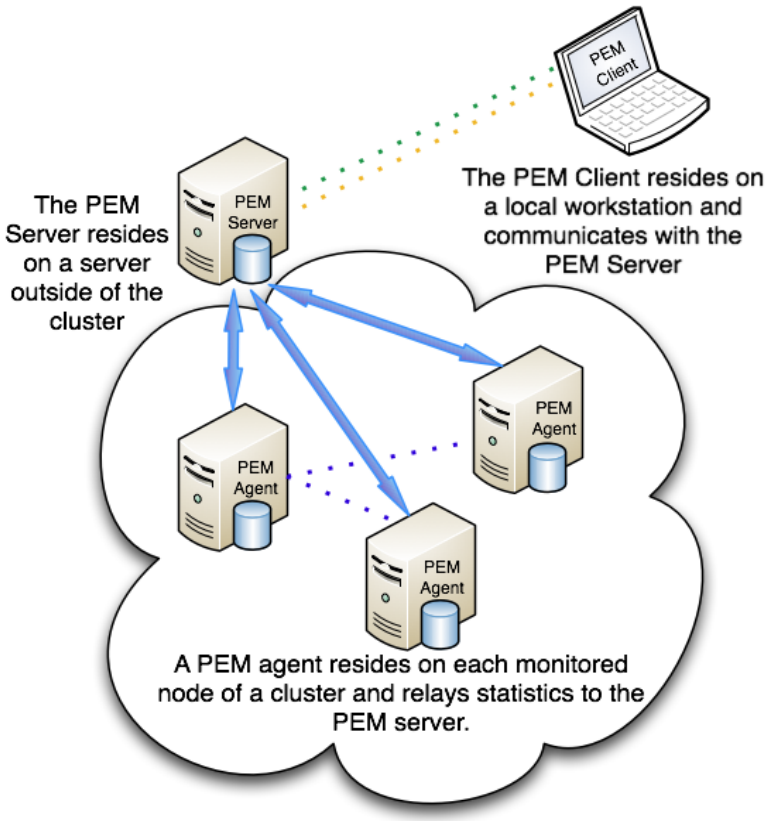


Figure 4.30 – A typical PEM installation.

After installing the PEM agent, the agent must be registered on *each* node that will be monitored by the PEM server. The steps that follow detail installing a PEM agent on an EDB Ark cluster, registering the agent with the server, and configuring the server to monitor the agent.

Step 1 – Modify a Database Engine Description to Install the PEM Agent

You must be an EDB Ark Administrative user to modify an engine description. Using Administrative credentials, connect to the Ark console, and navigate to the `Admin` tab. Select an engine `ID` from the list of engines in the `DB Engine Administration` list, and click `Edit Engine Details`. The `Edit Engine Details` dialog opens (see Figure 4.31).

Copyright © 2011 - 2017 EnterpriseDB Corporation. All rights reserved.

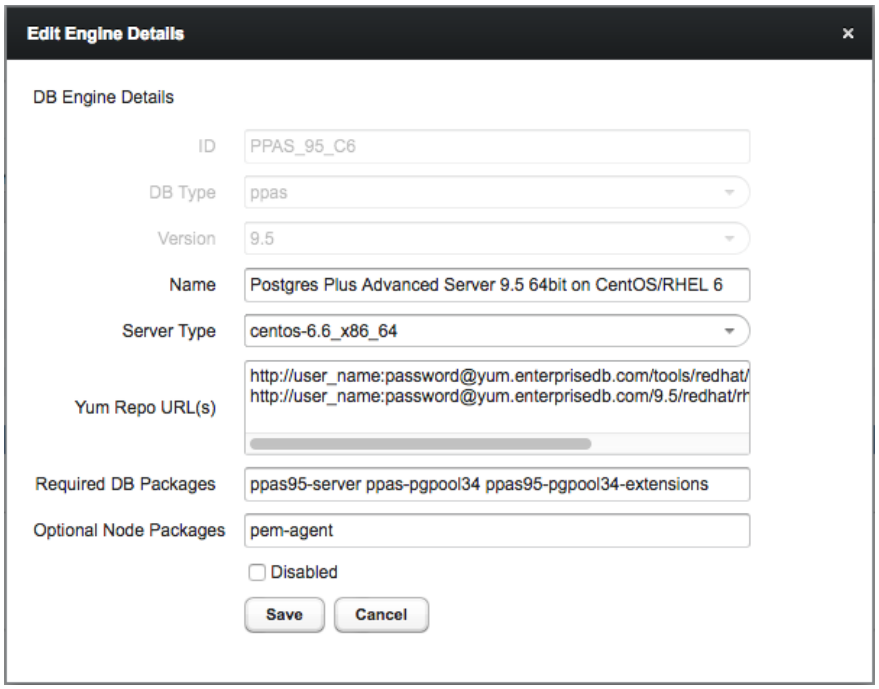


Figure 4.31 – Modifying the Engine Details dialog.

The name of the package that installs the PEM agent is named `pem-agent`. The package is distributed from the `enterprisedb tools` repository; by default, the `enterprisedb tools` repository is included in the `Yum Repo URL` field.

Add the name of the PEM agent RPM package (`pem-agent`) to the `Optional Node Packages` field on the `Edit Engine Details` dialog. Any EDB Ark clusters that are subsequently provisioned with that engine will automatically include an installation of the PEM agent on all nodes of the cluster. Please note that before monitoring a node, you must:

- modify the `pg_hba.conf` file on the node hosting the server to allow connections from any monitored node.
- modify the `pg_hba.conf` file on any monitored node, allowing connections from the PEM server.
- configure the agent on each monitored node.

The steps that follow provide detailed information about each configuration step.

For more information about administrative features of the EDB Ark console, please see the *EDB Ark Administrator's Guide*, available through the EDB Ark `Dashboard` tab.

Copyright © 2011 - 2017 EnterpriseDB Corporation. All rights reserved.

Step 2 – Create an EDB Ark Cluster

Navigate to the `Clusters` tab, and create a new cluster that is provisioned using the engine definition modified in `Step 1` (see Figure 4.32). As the cluster spins up, each node of the cluster will include the `pem-agent` RPM package.

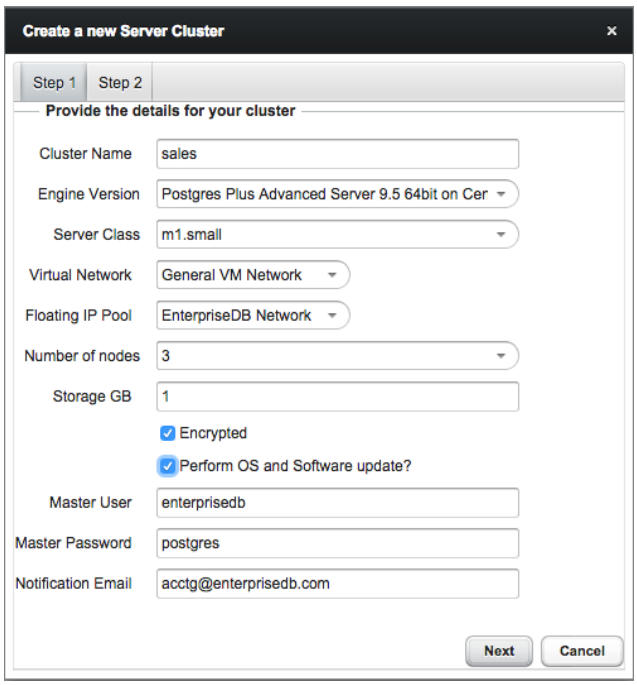


Figure 4.32 – Creating a new Server Cluster

For detailed information about creating a new server cluster, please see the *EDB Ark Getting Started Guide*, available through the EDB Ark `Dashboard` tab.

Please note: by default, a replica node on an OpenStack host does not have a public IP address. After creating the cluster, you must manually assign a public IP address to each node you wish to monitor with PEM. For more information, please see:

https://access.redhat.com/documentation/en/red-hat-enterprise-linux-openstack-platform/version-7/networking-guide/#configure_ip_addressing

Copyright © 2011 - 2017 EnterpriseDB Corporation. All rights reserved.

Step 3 – Modify the pg_hba.conf file to allow connections to the PEM Server

The PEM server consists of an instance of PostgreSQL, an associated PostgreSQL database for storage of monitoring data, and a server that provides web services for the PEM client. The PEM server may reside on a host outside of a monitored EDB Ark cluster, or on the master node of an Ark cluster.

Before a PEM agent that resides on an Ark cluster can communicate with the PEM server, you must modify the `pg_hba.conf` file (see Figure 4.33) of the PostgreSQL database that stores PEM statistics to allow connections from any monitored servers as well as the PEM client.

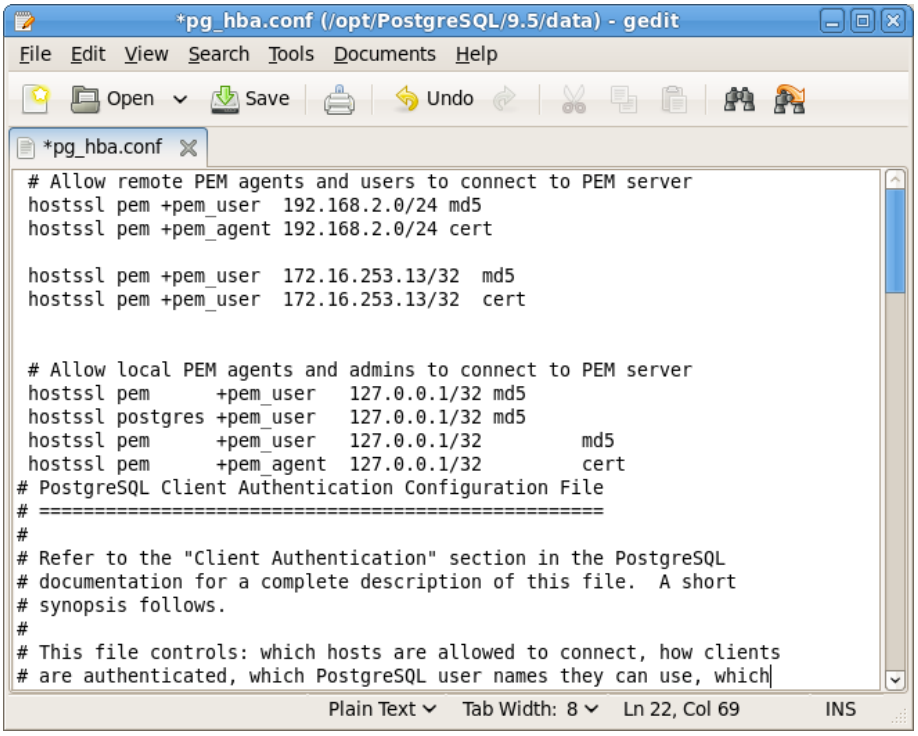


Figure 4.33 – Modifying the PEM Server's pg_hba.conf file.

With your choice of editor, modify the `pg_hba.conf` file (located by default in the `data` directory under the PostgreSQL installation), adding entries for the IP address of the EDB Ark cluster. The connection properties should allow connections that use `cert` and `md5` authentication.

For detailed information about modifying the `pg_hba.conf` file, please see the PostgreSQL documentation, available from the EnterpriseDB website at:

https://www.enterprisedb.com/resources/product-documentation

Copyright © 2011 - 2017 EnterpriseDB Corporation. All rights reserved.

Step 4 – Restart the PEM Server Database

After modifying the `pg_hba.conf` file for the PostgreSQL installation that stores statistical information for PEM, you must restart the PEM backing database server to apply the changes. The name of the PEM service is:

```
postgresql-9.x
```

Where *x* specifies the version. For example:

```
service postgresql-9.5 restart
```

Use the platform-specific command for your version to restart the PEM server.

Step 5 – Establish an SSH Session with the Monitored Node of the Ark Cluster

Use the `Download SSH Key` icon on the `Clusters` tab to download the SSH key for your cluster. When you download the key, a popup will open, informing you of the steps required to connect to the master node of your cluster (see Figure 4.34).

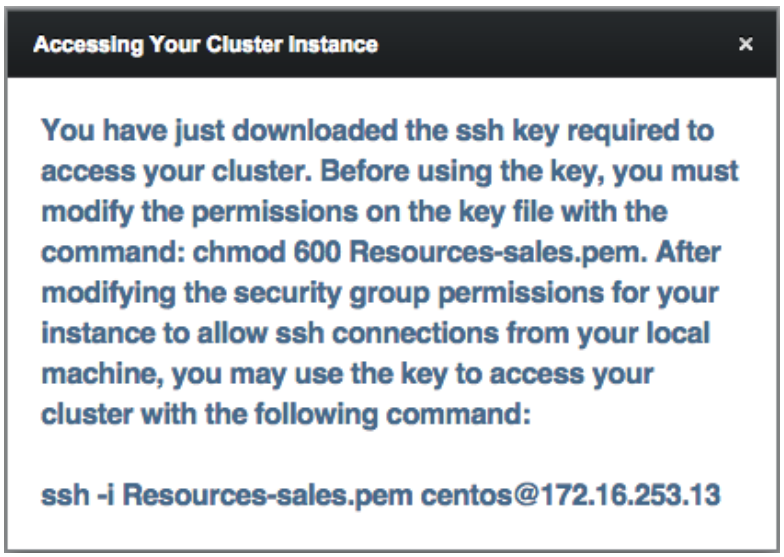


Figure 4.34 - Using SSH to connect to the Ark cluster.

Open a terminal window, modify the permissions on the downloaded file, and use the command shown on the popup to establish a connection with the server.

Copyright © 2011 - 2017 EnterpriseDB Corporation. All rights reserved.

Step 6 – Modify the pg_hba.conf file to Allow Connections from the PEM Server

Use your choice of editor to modify the `pg_hba.conf` file on the Ark node. By default, the `pg_hba.conf` file is located in `/var/lib/ppas/9.5/data`.

Add entries to the `pg_hba.conf` file that allow connections from the PEM server (see Figure 4.35).

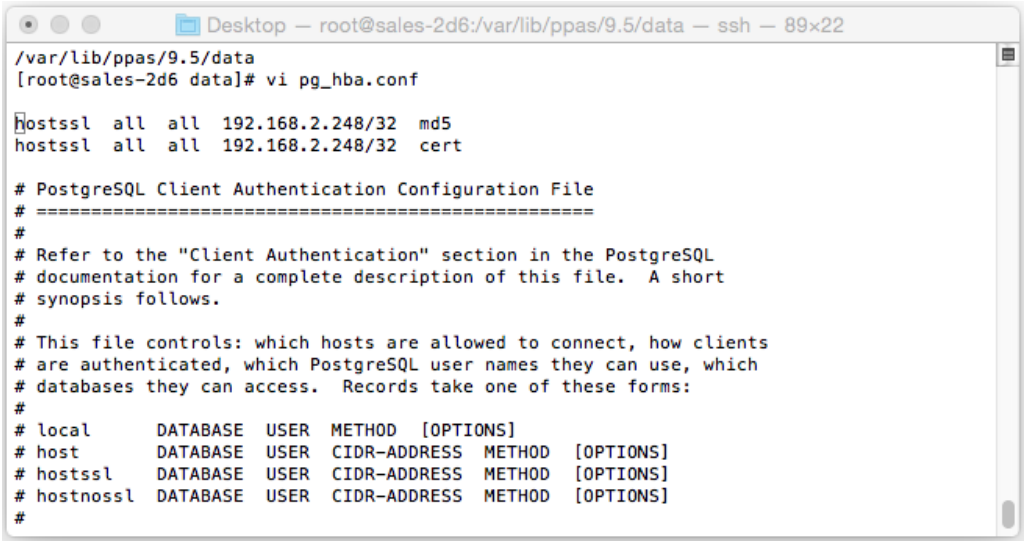


Figure 4.35 – Modifying the Ark cluster's pg_hba.conf file.

Step 7 – Restart the Database Server on the Ark Cluster

After modifying the `pg_hba.conf` file, you must restart the server to apply the changes. The name of the service is `Arkdb`. Use the platform and version specific command for your cluster to restart the `Arkdb` service.

Step 8 – Configuring the PEM Agent

You must register each PEM agent that resides in an Ark cluster with the PEM server. Using the SSH connection to the cluster node on which the agent resides, navigate into the directory that contains the PEM agent installation:

```
cd /usr/pem-6.0/bin
```

Then, invoke the PEM agent registration program:

Copyright © 2011 - 2017 EnterpriseDB Corporation. All rights reserved.

```
PGPASSWORD=password ./pemagent --register-agent --pem-
server x.x.x.x --pem-port port --pem-user user_name
```

Where:

x.x.x.x specifies the IP address of the PEM server.

port specifies the port on which the server is listening for connections

user_name specifies the name of the PEM user.

The program will confirm that the agent was registered successfully (see Figure 4.36).

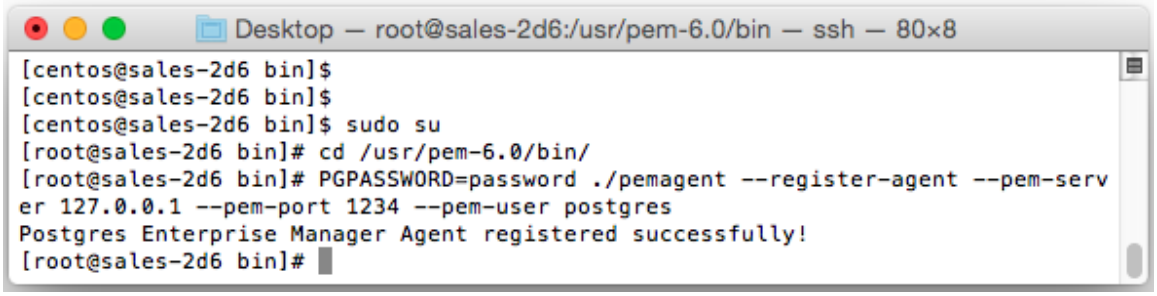


Figure 4.36 – Registering the PEM agent.

After registering the agent, use the following command to ensure that the service is configured to restart when if the node restarts, and that the pemagent service is running:

```
chkconfig pemagent on && service pemagent start
```

For more information about Postgres Enterprise Manager, and to download PEM documentation, please visit the EnterpriseDB website at:

https://www.enterprisedb.com/products/edb-postgres-platform/edb-postgres-enterprise-managerpem

Copyright © 2011 - 2017 EnterpriseDB Corporation. All rights reserved.

4.1.4 User Administration

Options in the `User Administration` section of the `Admin` tab allow an administrative user to access a list of connected users or to display a message to all connected users (see Figure 4.37).

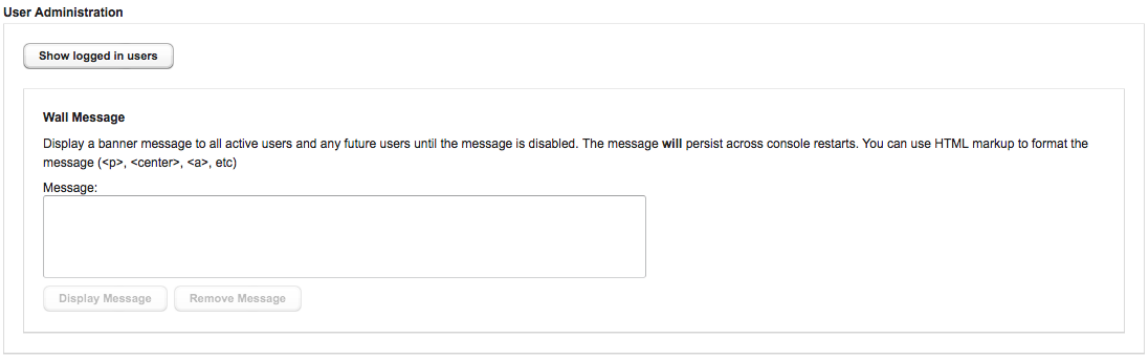


Figure 4.37 –User administration features of the OpenStack console.

Click the `Show logged in users` button to display the `Logged in users` dialog (see Figure 4.38).

Figure 4.38 – 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.

Copyright © 2011 - 2017 EnterpriseDB Corporation. All rights reserved.

Provide a message in the `Message` field (shown in Figure 4.39) and click the `Display Message` button to add an announcement to the top of the user console. A message may include HTML tags to control the displayed format, and will wrap if the message exceeds the width of the screen.

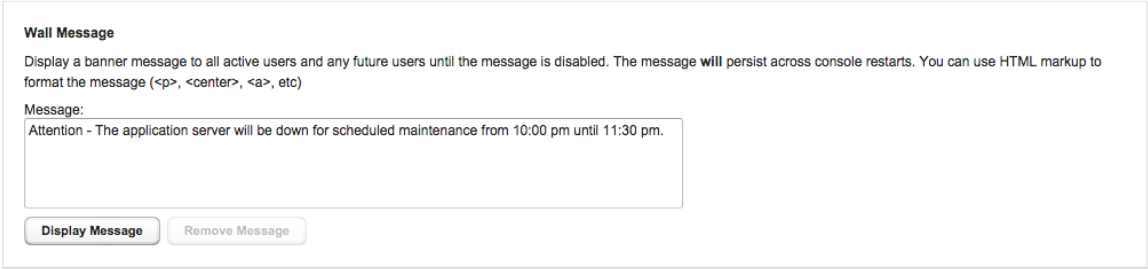


Figure 4.39 - Modifying the Wall Message.

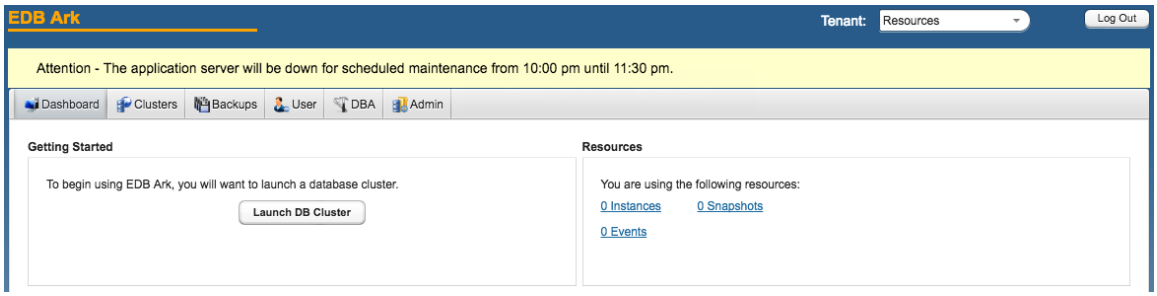


Figure 4.40 - Displaying a wall message.

The console may take a few seconds to refresh. Once processed by the server, the message will be displayed to console users when their screens refresh (see Figure 4.40).

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

Copyright © 2011 - 2017 EnterpriseDB Corporation. All rights reserved.

4.1.4.1 User Management Features on an Amazon Host

If your console resides on an Amazon AMI, the Ark administrative console provides a user management interface that allows you to:

- review user information.
- create and manage users.
- delete users.
- delete user-owned properties such as clusters, snapshots and keypairs.

A table with detailed information about each user is displayed in the `User Administration` section of the `Admin` tab (see Figure 4.41).

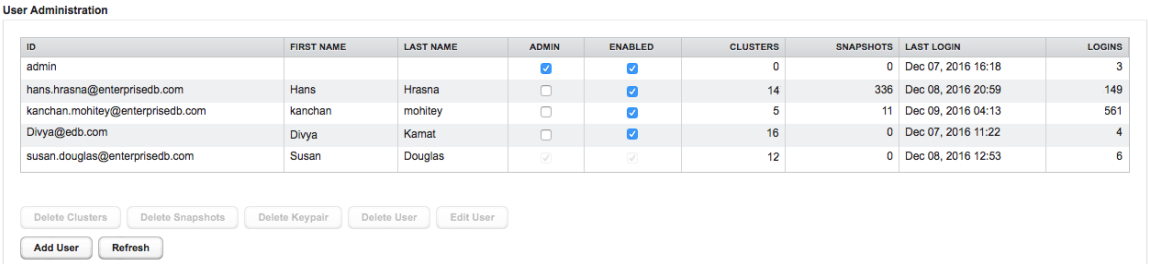


Figure 4.41 – The user table of an AWS console

Columns within the table provide information about the current AWS console users:

- The user's login name is displayed in the `ID` column.
- The user's first name is displayed in the `FIRST NAME` column.
- The user's last name is displayed in the `LAST NAME` column.
- If the user has administrative access to the console, the `ADMIN` column displays a blue check mark.
- If the user account is currently active (the user can log in), the `ENABLED` column displays a blue check mark.
- The number of clusters currently owned by the user is displayed in the `CLUSTERS` column.
- The number of cluster snapshots owned by the user is displayed in the `SNAPSHOTS` column.
- The date and time of the last login is displayed in the `LAST LOGIN` column. The time zone displayed is based on the time zone used by the operating system.

Copyright © 2011 - 2017 EnterpriseDB Corporation. All rights reserved.

- The `LOGINS` column displays a cumulative total of the number of times that the user has logged in.

Use the buttons below the AWS user table to manage user accounts for the AWS console and user-owned objects.

Adding a User on an AWS Console

Click the `Add User` button to access the `Add User` dialog (see Figure 4.42) and create a new user account for the current console.

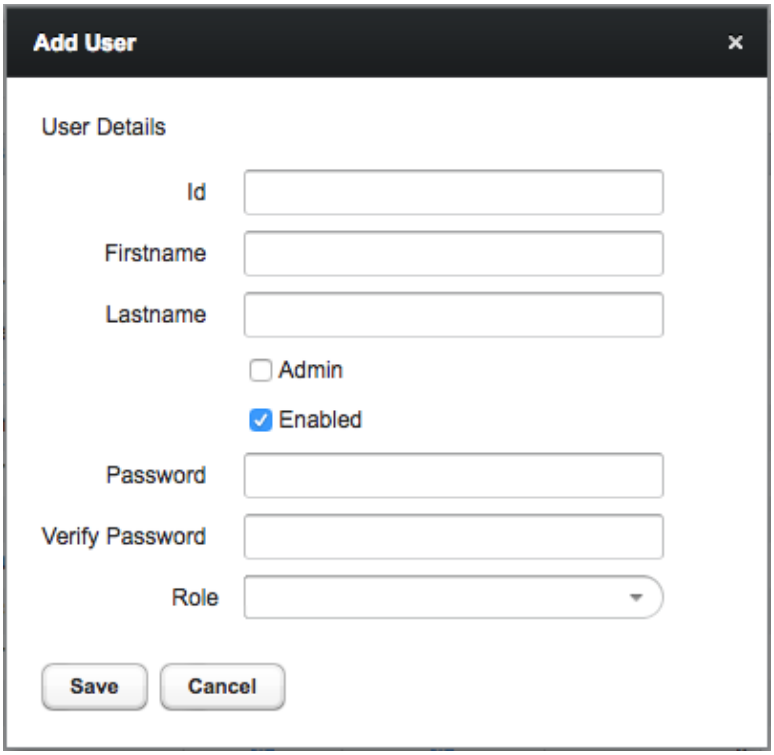


Figure 4.42 – The Add User dialog.

Provide information about the new user account:

- Use the `Login` field to provide the identifier that the user will provide when logging in to the console; each identifier must be unique.
- Provide the user's first name in the `First Name` field.
- Provide the user's last name in the `Last Name` field.
- To allow the user administrative access to the Ark console, check the box next to `Admin`.

Copyright © 2011 - 2017 EnterpriseDB Corporation. All rights reserved.

- Check the box next to `Enabled` if the user should be allowed to log in to the console.

- Provide a password associated with the user account in the `Password` field.

- Confirm the password in the `Verify Password` field.

- Select a previously defined Amazon role ARN from the drop-down list in the `Role` field, or copy a different role ARN into the field.

  The role ARN must be defined on the AWS console by an Amazon administrator. Each role will be able to access all clusters that are created by users that share the common role ARN. To create an isolated user environment, a user must have a unique Amazon role ARN.

  If you copy an Amazon role ARN into the `Role` field, a popup will open, prompting you for the AWS `ExternalId` associated with the user. To locate the `ExternalId`, connect to the Amazon management console, and navigate to the `IAM Roles` page. Select the role name from the list, and then click `Trust Relationships` tab. The `ExternalId` associated with the Role ARN is displayed in the `Conditions` section of the `Summary` page.

Modifying a User on an AWS Console

Click the `Edit User` button to open a dialog that allows you to modify user properties for the user that is currently highlighted in the user table (see Figure 4.43).

Copyright © 2011 - 2017 EnterpriseDB Corporation. All rights reserved.

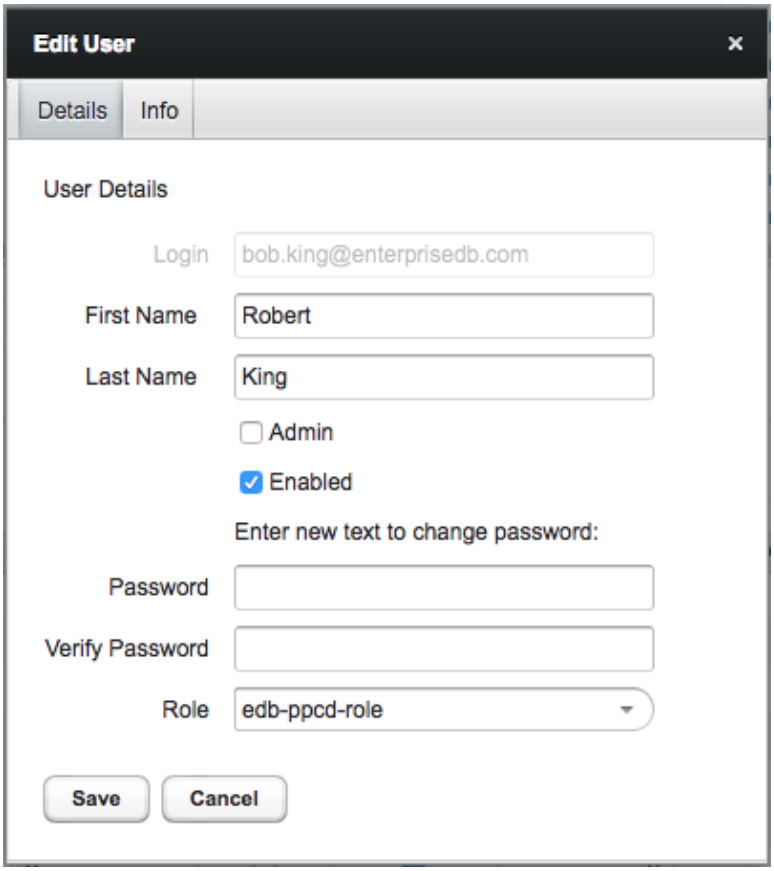


Figure 4.43 – The Edit User dialog.

Deleting Users and Objects on an AWS Console

If the backing host of your Ark console is Amazon AWS, you can use features on the `Admin` tab to delete objects that belong to a user, and the user account. Highlight a user name in the `User Administration` table, and click:

- The `Delete Clusters` button to delete all clusters that belong to the selected user.
- The `Delete Snapshots` button to delete any cluster backups that belong to the selected user.
- The `Delete Keypair` button to delete the SSH keypair associated with the selected user account.

After deleting the objects owned by a user, you can use the `Delete User` button to remove the user account. To delete a user, highlight the name of a user in the user table, and click the `Delete User` button.

Copyright © 2011 - 2017 EnterpriseDB Corporation. All rights reserved.

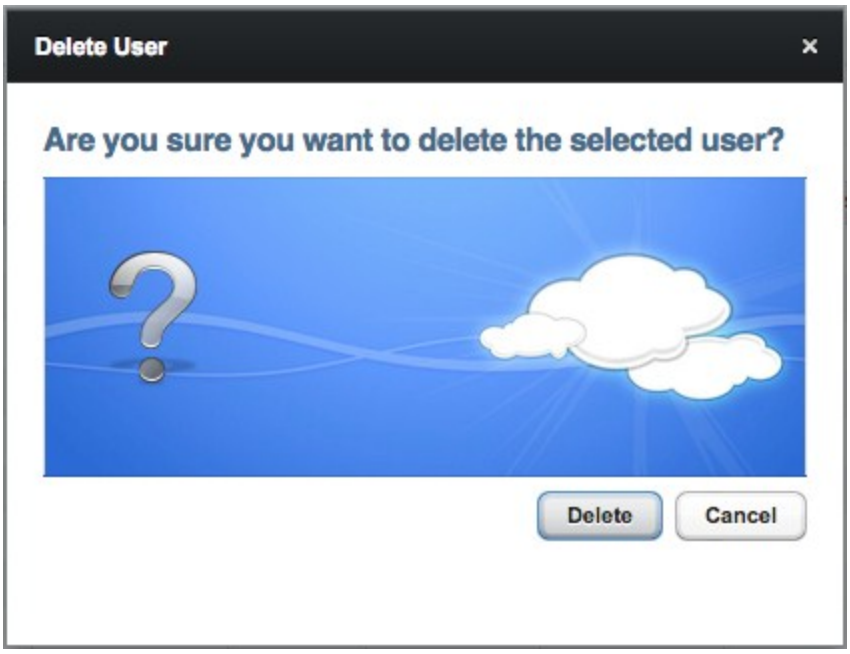


Figure 4.44 – The Delete User dialog.

The Ark console will open a popup, asking you to confirm that you wish to delete the selected user (see Figure 4.44). Click `Delete` to remove the user account, or `Cancel` to exit the popup without deleting the account.

4.1.5 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 (see Figure 4.45).

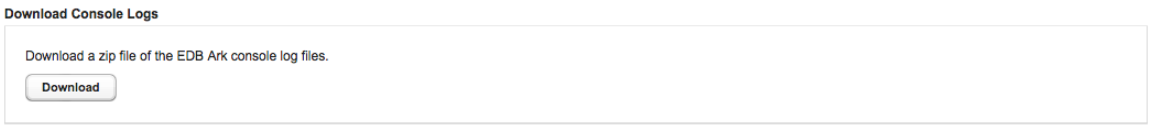


Figure 4.45 – The user table of an AWS console.

You can also review the console logs via an ssh session; server log files are stored in:

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

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

Copyright © 2011 - 2017 EnterpriseDB Corporation. All rights reserved.

```
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 4.46) 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.

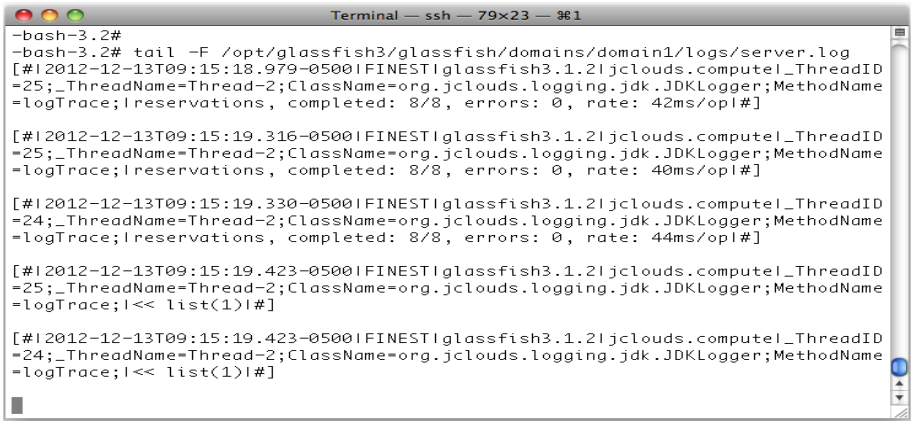


Figure 4.46 - Following the log file with the tail utility.

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
```

Copyright © 2011 - 2017 EnterpriseDB Corporation. All rights reserved.

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 4.47) is accessible only to administrative users.

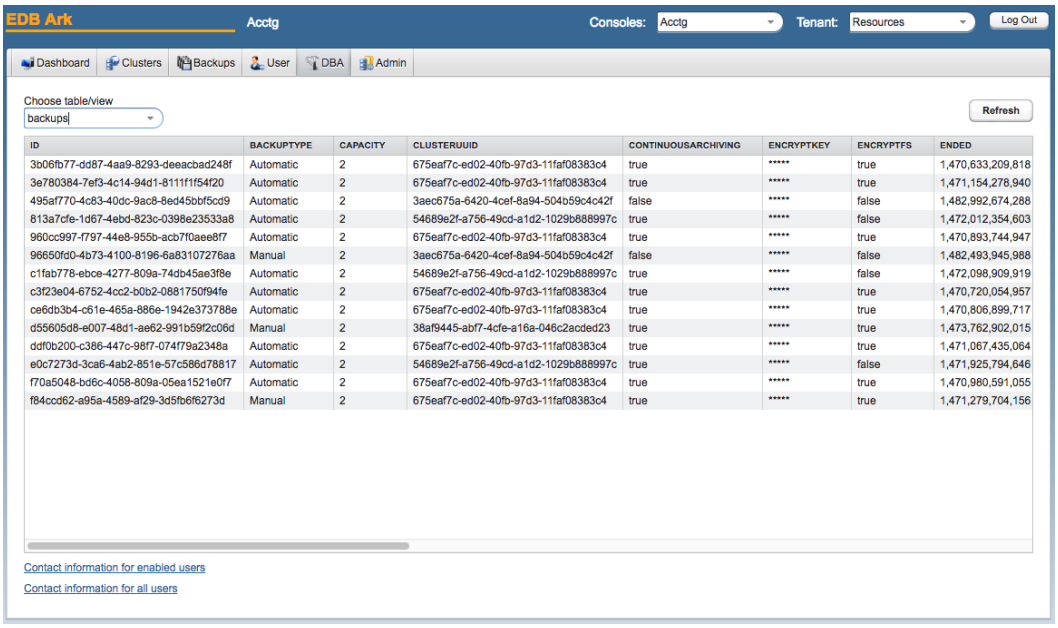


Figure 4.47 - The DBA tab.

Use the `Choose table/view` drop down listbox (shown in Figure 4.48) to select a view.

Figure 4.48 - The table/view listbox.

Copyright © 2011 - 2017 EnterpriseDB Corporation. All rights reserved.

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 4.49) to download a comma-delimited list of users and user information.

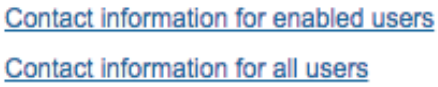


Figure 4.49 - The contact information links

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).

Copyright © 2011 - 2017 EnterpriseDB Corporation. All rights reserved.

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 activation 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.
IOPS	The IOPs value for the volume.
OPTIMIZED	True if the cluster is optimized, False if the cluster is not optimized.

Copyright © 2011 - 2017 EnterpriseDB Corporation. All rights reserved.

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 `(encrypted)`.
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.
CONTINUOUSARCHIVING	True if archiving is enabled; false if archiving is disabled.
CLUSTERUUID	The identifier of the cluster from which the backup was created.
XLOGLOCATION	The location of the Xlog file for the backup.
XLOGFILENAME	The identifier of the Xlog file for the backup.
WALARCHIVECONTAINER	The name of the archive container in which the WAL logs are stored.
ENCRYPTFS	True if the content of a backup is stored on an encrypted file system; false if they are not.
ENCRYPTKEY	The key associated with the backup (obscured).
TENANT	The tenant in which the cluster resides.
YUMUPDATE	True if updates are enabled for the cluster; false if they are not.
DBENGINE_ID	The engine number of the database engine used by the cluster.

4.3.4 consoleurl

The `consoleurl` table provides a list of the resources currently made available by the console switcher.

Column Name	Description
ID	The row ID.
NAME	The name of the cluster that resides on th eURL.
URL	The URL of the master node of the cluster.

Copyright © 2011 - 2017 EnterpriseDB Corporation. All rights reserved.

4.3.5 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	`true` if the engine is no longer supported; `false` 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.

4.3.6 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	`true` if auto-scaling is enabled on the cluster; `false` 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.
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`

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.
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.

Copyright © 2011 - 2017 EnterpriseDB Corporation. All rights reserved.

Column Name	Description
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 (edb or postgres).
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 thedatabase 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.
INSTANCESTATE	The current state of the node.
MASTERPW	The password of the cluster owner.
MASTERUSER	The name of the cluster owner.
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 id 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.
OPTIMIZED	Boolean value; true if an instance is optimized; false if not (valid for AWS users only).
IOPS	The requested IOPS setting for the cluster (valid for AWS users only).
MONITORINGLB	Boolean value; true if load balancing is enabled, false if load balancing is not enabled.
CASTATE	The most-recent continuous archiving state of the instance.
CONTINUOUSARCHIVING	Boolean value; true if continuous archiving is enabled, false if continuous archiving is not enabled.
CLUSTERUUID	The unique cluster identifier.
VPC	The VPC ID (valid for AWS users only).
ENCRYPTFS	True if encryption is enabled for the cluster; false if it is not.
ENCRYPTKEY	The encryption key for the cluster.

Copyright © 2011 - 2017 EnterpriseDB Corporation. All rights reserved.

Column Name	Description
CLUSTERKEY	The SSH key shared by all of the instances in the cluster.
CLUSTERKEYNAME	The name of the SSH key.
IPPOOL	The name of the floating IP pool (valid for OpenStack users only).
LBPORT	The load balancing port of the instance.
NOTIFICATIONEMAIL	The notification email for the cluster.
TENANT	The tenant in which the node was created.
VERSION_NUM	The version of EDB Ark under which the instance was created.
VOLUMETYPE	If supported, the volume type of the cluster.
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.
DBENGINE_ID	The selected database engine installed on the instance.

4.3.7 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 nodestatistics 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).

4.3.8 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 pcshistory 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.

Copyright © 2011 - 2017 EnterpriseDB Corporation. All rights reserved.

4.3.9 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.

4.3.10 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.

4.3.11 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 `backups` 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.

Copyright © 2011 - 2017 EnterpriseDB Corporation. All rights reserved.

5 Securing EDB Ark

Each cluster has an associated AWS or 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 `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.

Copyright © 2011 - 2017 EnterpriseDB Corporation. All rights reserved.

5.1 Modifying a Security Group for an OpenStack Hosted Console

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).

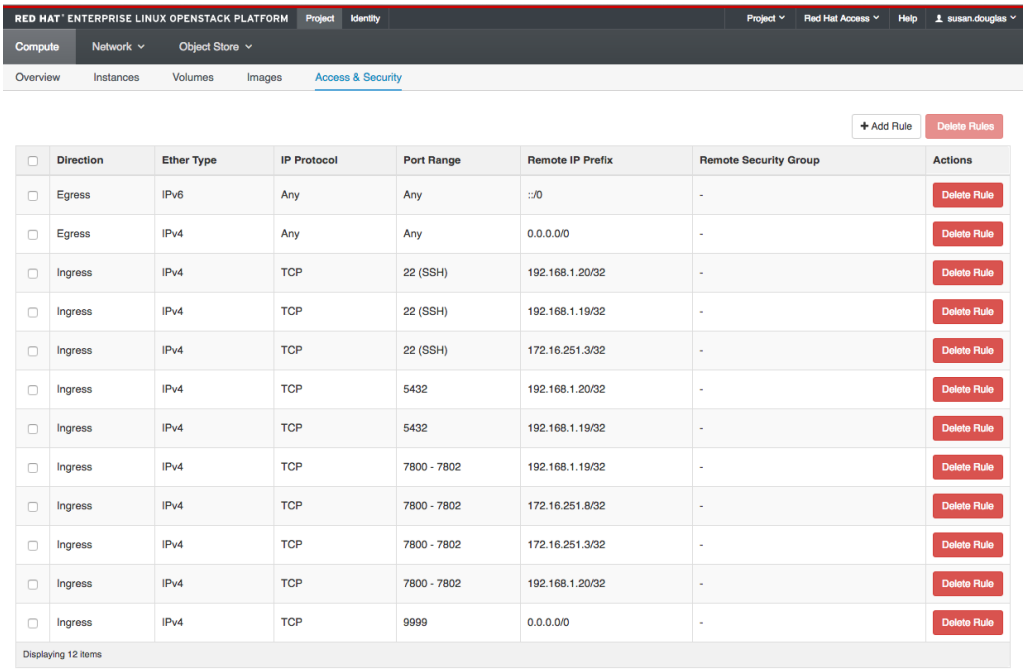


Figure 5.1 – Detailed security rules for a cluster.

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`.

Copyright © 2011 - 2017 EnterpriseDB Corporation. All rights reserved.

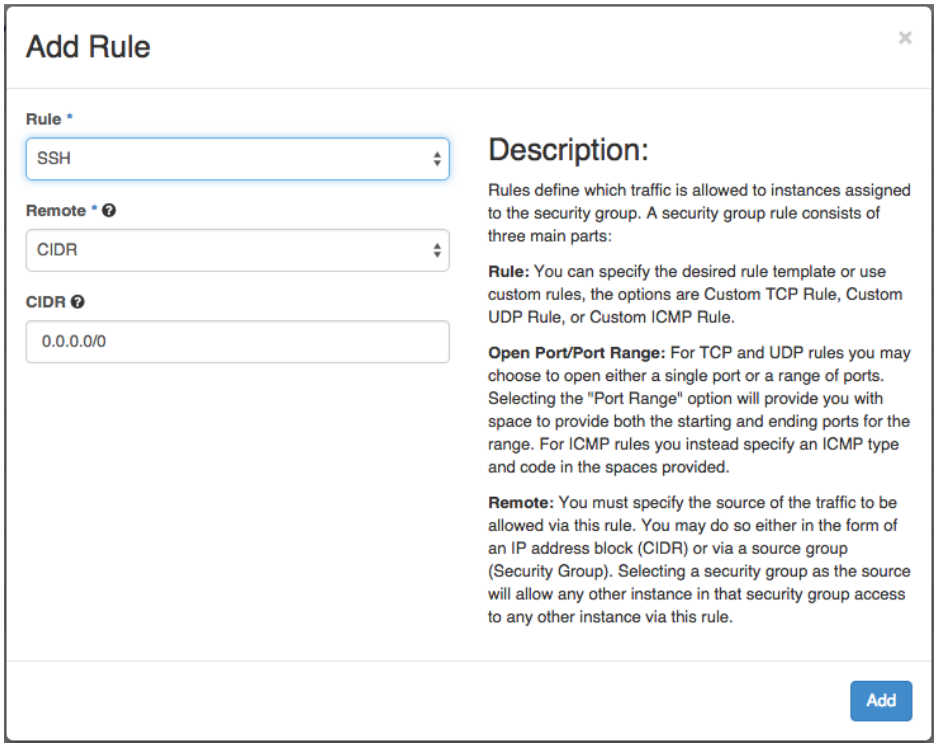


Figure 5.2 – Opening a port for an SSH connection.

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`.

- Use the `CIDR` field to 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/9.6/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.

Copyright © 2011 - 2017 EnterpriseDB Corporation. All rights reserved.

5.2 Modifying a Security Group for an Amazon AWS Hosted Console

Security groups for Ark clusters that reside on an AWS host are managed through the Amazon management console; Amazon administrative privileges are required to review or modify the security group entries.

To manage a security group for a cluster, connect to the AWS management console, and locate the cluster on the `Instances` dashboard. Highlight the cluster name, and scroll through the columns to the right. Click the name of the security group (in the `Security Groups` column) to review detailed information about the rules that are currently defined for the cluster.

To modify a security group and add a rule that allows connections from an outside client (such as ssh), navigate to the `Inbound` tab, and click the `Edit` button. When the `Edit inbound rules` dialog opens, click the `Add Rule` button to add a new line to the list of rules (see Figure 5.3).

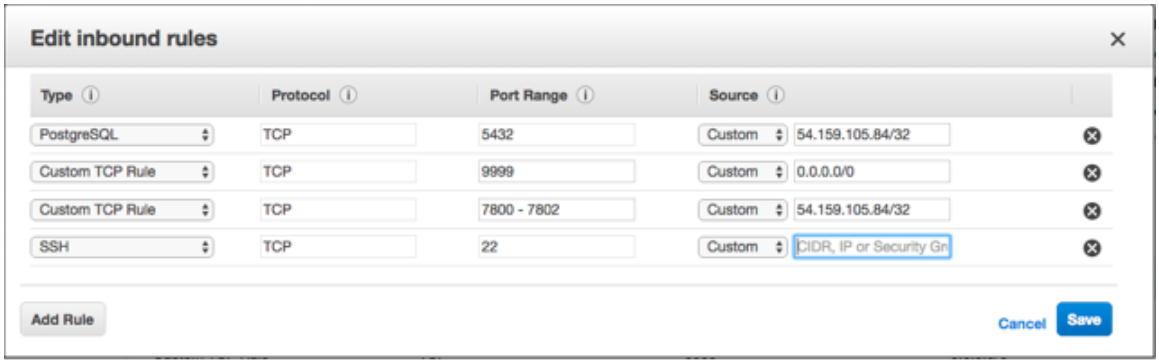


Figure 5.3 – Opening a port for an SSH connection.

Specify the rule type, the protocol type, the port (or port range) on which inbound connections will be accepted, and the CIDR-formatted address from which you will be connecting.

For detailed information about specifying a CIDR address, see:

http://www.postgresql.org/docs/9.6/static/datatype-net-types.html

When you've defined the rule, click `Save` to add the entry to the inbound rules list.

Please consult the Amazon documentation for detailed information about managing the security group for a virtual private cloud:

http://docs.aws.amazon.com/AmazonVPC/latest/UserGuide/VPC_SecurityGroups.html

Copyright © 2011 - 2017 EnterpriseDB Corporation. All rights reserved.

5.3 Using ssh to Access a Server

EDB Ark creates an `ssh` key when you create a new cluster; each cluster has a unique key. Before connecting to a Postgres instance that resides on the cloud via an `ssh` encrypted connection, you must download the `ssh` key, and adjust the privileges on the key file.

To download your private key, navigate to the `Clusters` tab, and click the `Download SSH Key` icon. The `Accessing Your Cluster Instance` popup opens (see Figure 3.28).

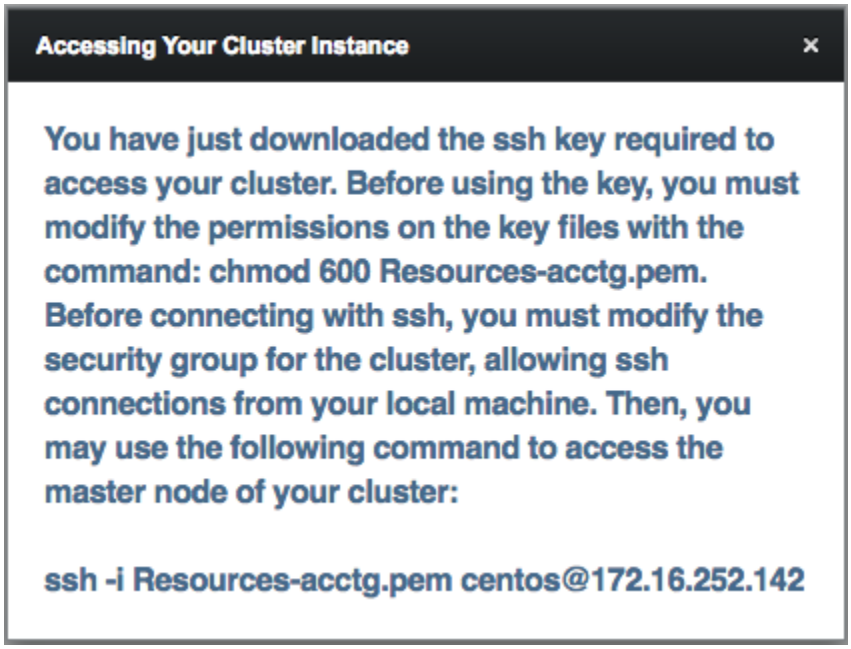


Figure 3.28 – Accessing Your Cluster Instance.

The popup displays the tenant name, the cluster name, the name that you should use when connecting to the cluster, and the IP address to which you should connect.

Before using the private key, you must modify the permissions on the keyfile. Use the following command to restrict file permissions:

```
chmod 0600 ssh_key_file.pem
```

Where *`ssh_key_file.pem`* specifies the complete path and name of the EDB Ark `ssh` private key file.

Copyright © 2011 - 2017 EnterpriseDB Corporation. All rights reserved.

After modifying the key file permissions, you can use `ssh` to connect to the cluster. Include the complete path to the key file when invoking the command provided on the `Accessing Your Cluster Instance` popup.

Please note: Postgres Server applications must be invoked by the Postgres cluster owner (identified when creating an EDB Ark cluster as the `Master User`). If you are using a PostgreSQL server, the default user name is `postgres`; if you are using Advanced Server, the default user name is `enterprisedb`. To change your identity after connecting via `ssh`, use the `su` command:

```
# sudo su database_user_name
```

5.4 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
```

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
```

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
```

Copyright © 2011 - 2017 EnterpriseDB Corporation. All rights reserved.

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.

5.5 Post-Installation Recommendations

SE Linux

Copyright © 2011 - 2017 EnterpriseDB Corporation. All rights reserved.

During the installation process, SE Linux is disabled on the console host. Please note that SE Linux must remain disabled for the Ark console and clusters to function properly.

Create a Secondary User Account

The Ark console installation process creates an administrative user (named `centos`) with ssh access to the console host. After installing the Ark console, you should use ssh to connect to the console host, and create a secondary user account that can be used to login and gain `root` privileges in the event that the `centos` user should lose ssh access for any reason.

Copyright © 2011 - 2017 EnterpriseDB Corporation. All rights reserved.

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.

Copyright © 2011 - 2017 EnterpriseDB Corporation. All rights reserved.

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 4.1.4.

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
   ```

Copyright © 2011 - 2017 EnterpriseDB Corporation. All rights reserved.

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.

   ```
   [root@edb-ark-test ppcd]# ./postInstall.sh
   Script will upgrade the application! Is the EDB-ARK console
   in a steady state (no logged in users, no activity in the
   console)?
   The following files were in conflict during the last yum
   update and need to be either removed or merged with the
   existing files.
   /var/ppcd/PPCDConsole/WEB-
   INF/classes/i18n.properties.rpmnew
   /var/ppcd/PPCDConsole/VAADIN/themes/pcsconsole/jsppage.css.
   rpmnew
   /var/ppcd/PPCDConsole/VAADIN/themes/pcsconsole/styles.css.r
   pmnew
   Are you sure you want to continue? <y/N> y

         Updating EDB-ARK Application...
   ```

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

During the upgrade process, the Ark RPM is careful not to overwrite any existing files that have been modified. The package identifies any pre-existing files, and creates the new (potential replacement) files with the `.rpmnew` extension.

When the `yum update` completes, you should examine any files with the `.rpmnew` extension to see if any functionality (such as new parameters) should be merged into your current files, and then delete the file with the `.rpmnew` extension. The `./postInstall.sh` script (invoked in Step 6) will provide a list of any files that were in conflict.

Copyright © 2011 - 2017 EnterpriseDB Corporation. All rights reserved.

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:

```
/var/ppcd/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:

```
/var/ppcd/PPCDConsole/VAADIN/themes/pcsconsole/images
```

Copyright © 2011 - 2017 EnterpriseDB Corporation. All rights reserved.

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:

```
/var/ppcd/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:

```
/var/ppcd/PPCDConsole/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.

Copyright © 2011 - 2017 EnterpriseDB Corporation. All rights reserved.

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: `0f42d1934a1a19f3d25d6288f2a3272c6143fc5d`

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
```

Copyright © 2011 - 2017 EnterpriseDB Corporation. All rights reserved.

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
```

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/9.6/static/libpq-pgpass.html

Modifying the GlassFish Console Password

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

name: `admin`

Copyright © 2011 - 2017 EnterpriseDB Corporation. All rights reserved.

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. 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.
```

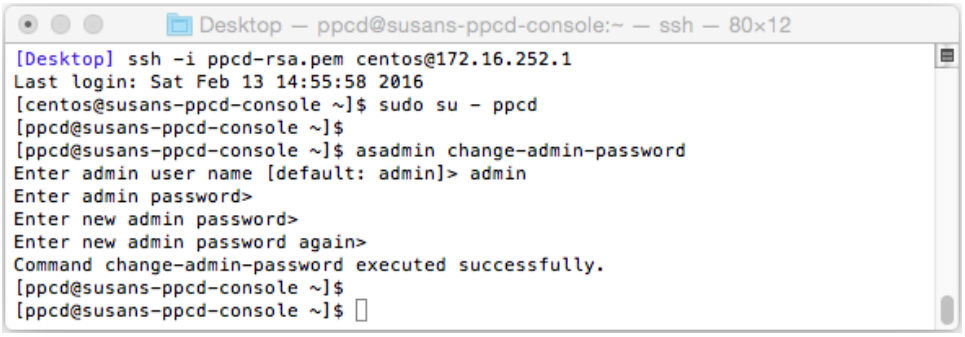


Figure 6.1 – Changing the console user's password.

If you are 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:

Copyright © 2011 - 2017 EnterpriseDB Corporation. All rights reserved.

```
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.
```

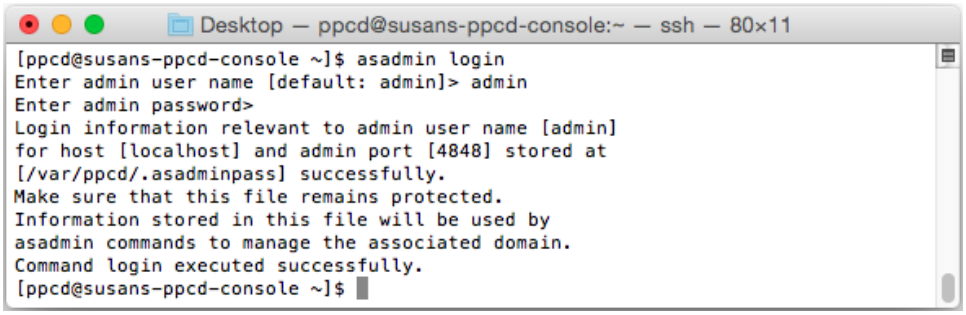


Figure 6.1 – Invoking asadmin login.

Copyright © 2011 - 2017 EnterpriseDB Corporation. All rights reserved.

7 Recovering From a Console Failure

User and instance information used by the Ark console is stored in tables in a `postgres` database. If the console application should fail, the information will persist in the console database, and will be available when the console application restarts.

If the system hosting the application database fails, then all information about the console database and registered users will be lost unless you have retained a backup.

Parameters in the `ppcd.properties` file configure the EDB Ark backup script to take automatic backups of the console database after the registration of each new user, and hourly. If you do not wish to use the Ark backup script to implement backups, you should maintain regular backups of your console database.

If you wish to manually save backups, you can use the Postgres `pg_dump` or `pg_dumpall` command to archive the console database. Then, you can then use the `pg_restore` command to restore the console database if necessary.

7.1 Enabling Console Backups with the EDB Ark Backup 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= 0f42d1934a1a19f3d25d6288f2a3272c6143fc5d
# DB name to connect to
# console.db.name=postgres
```

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

Copyright © 2011 - 2017 EnterpriseDB Corporation. All rights reserved.

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 a directory to which backups will be written. Please note that you must create the directory specified. The `ppcd` user must have sufficient privileges to write to the specified directory.

The backup directory specified shound 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
```

On an Amazon hosted console, you can use the `console.db.backup.container` and `console.db.backup.folder` parameters to specify the name of a container (an Amazon S3 bucket) in which console backups will be stored, and a console-specific folder name. If no value is specified for `console.db.backup.folder`, the value will default to `default`.

```
# Optional bucket name in which to store console backups
# console.db.backup.container=

# Unique name for the console backup folder that identifies this
# console, i.e. 'dev.console'. Default name is 'default'
# console.db.backup.folder=
```

Copyright © 2011 - 2017 EnterpriseDB Corporation. All rights reserved.

7.1.1 Recovering the Console from a Backup Script

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.

7.1.2 Using the Recover Option on an AWS Backed Console

If the console is configured to support console backups, the Ark console will display the `Register or Recover` dialog (as shown in Figure 7.1) if the console cannot locate a registered user.

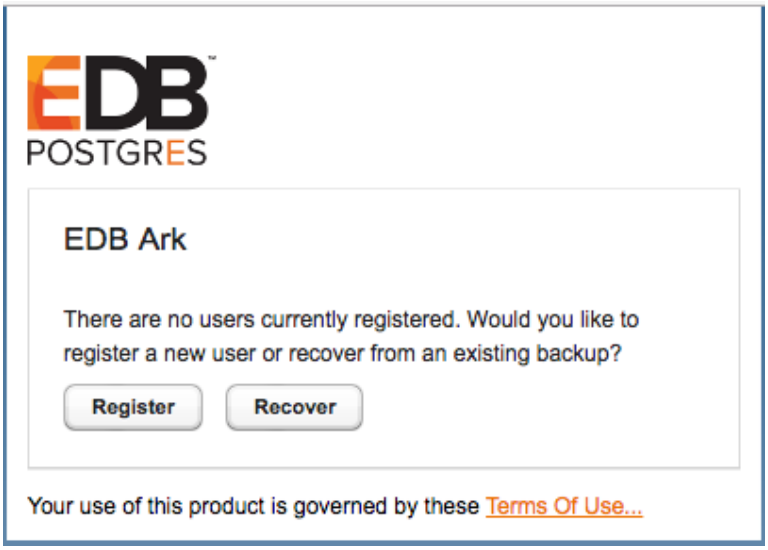


Figure 7.1 - The connection dialog.

Select `Recover` to instruct the Ark console to use the backup of your old console when starting. The recovered console will contain the previous list of registered users, monitoring data, and events from the last time that the database was backed up.

Copyright © 2011 - 2017 EnterpriseDB Corporation. All rights reserved.

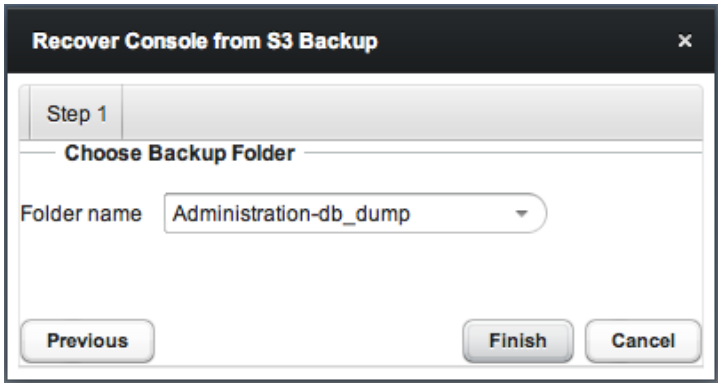


Figure 7.3 - The Step 2 tab of the Recover Console dialog.

The `Folder name` drop-down listbox (see Figure 7.2) will contain a list of backup sources available for use during the recovery. Select a backup source, and click `Finish` to instruct Ark to reload the backup, or `Cancel` to exit without recovering the console.

Please note: To successfully recover, the selected folder must contain a backup of your console database. If you do not have a backup, the recovery attempt will return an error message.

Copyright © 2011 - 2017 EnterpriseDB Corporation. All rights reserved.

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: *`script_output`*.
Database State Changed to *`db_state`*	The `MASTER|REPLICA` database server *`dns_name`* in cluster *`cluster_name`* is now `STOPPED|STARTING|RUNNING|WARNING|UNKNOWN` in location *`availability_zone`*.
Load Balancer Port Error	The `MASTER|REPLICA` database server *`dns_name`* in cluster *`cluster_name`* in location *`availability_zone`* is reporting an error determining the load balancer port.
Load Balancer Port Notification	The `MASTER|REPLICA` database server *`dns_name`* in cluster *`cluster_name`* is now `RUNNING|STARTING|STOPPED|WARNING|UNKNOWN` in location *`availability_zone`* using port *`port_number`*.
Continuous Archiving State Changed to *`db_state`*	Continuous Archiving on the `master|replica` database server *`dns_name`* in cluster *`cluster_name`* is operating normally.
Continuous Archiving State Changed to *`db_state`*	A problem was detected with continuous archiving on the `master|replica` database server *`dns_name`* in cluster *`cluster_name`*.
Data Storage Scaling *`cluster_name`*	Data storage is being added to cluster *`cluster_name`* because the auto-scaling threshold was reached.

Copyright © 2011 - 2017 EnterpriseDB Corporation. All rights reserved.

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

Copyright © 2011 - 2017 EnterpriseDB Corporation. All rights reserved.

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 *cluster_name* completed.	The termination of cluster *cluster_name* has completed.
WARNING: Termination Protection *instance_id*.	The system was not able to terminate instance {0} in cluster *cluster_name* because termination protection is enabled. You must disable termination protection before this instance can be terminated.
OS/SW update PASSED on node *instance_id.*	Yum update results for node: *dns_name* Yum exit status: *exit_status* You may also consult the yum log on the node (usually in `/var/log/yum.log`) 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 *instance_id.*	Yum update results for node: *dns_name* Yum exit status: *exit_status* You may also consult the yum log on the node (usually in `/var/log/yum.log`) 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: *status*	The OS/SW status on node *dns_name* of cluster *cluster_name* 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: *status*	The OS/SW status on node *dns_name* of cluster *cluster_name* 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 *dns_name* and run "`yum --security check-update`" to diagnose the issue manually.
Unable to delete Security Group *group_name*.	The system was not able to delete the Security Group named *group_name* in cluster *cluster_name*. This could be because one or more instances in the cluster could not be terminated. This Security Group will need to be manually

Copyright © 2011 - 2017 EnterpriseDB Corporation. All rights reserved.

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

Copyright © 2011 - 2017 EnterpriseDB Corporation. All rights reserved.

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
```

Copyright © 2011 - 2017 EnterpriseDB Corporation. All rights reserved.

10 AWS Policies

10.1 Reference - AWS Service User Security Policy

When you define an Amazon service user, you are required to provide an inline security policy. You can use the following security policy when registering a service user:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "Stmt1389628412000",
            "Effect": "Allow",
            "Action": [
                "sts:GetFederationToken",
                "sts:AssumeRole"
            ],
            "Resource": [
                "*"
            ]
        }
    ]
}
```

Copyright © 2011 - 2017 EnterpriseDB Corporation. All rights reserved.

10.2 Reference – AWS Service Role Security Policy and Trust Relationship

When you define an Amazon service role, you are required to provide a security policy and an updated trust relationship policy document. You can use the following trust relationship policy document:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "",
      "Effect": "Allow",
      "Principal": {
        "Service": "ec2.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    },
    {
      "Sid": "",
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::your_account_number:root"
      },
      "Action": "sts:AssumeRole",
      "Condition": {
        "StringEquals": {
          "sts:ExternalId": "EDB-ARK-SERVICE"
        }
      }
    }
  ]
}
```

You can use the following security policy when registering a service user:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "s3:*",
            "Resource": "arn:aws:s3:::*"
        }
    ]
}
```

Copyright © 2011 - 2017 EnterpriseDB Corporation. All rights reserved.

10.3 Reference – AWS User Security Policy

When you define an Amazon role, you are required to provide a security policy. The following text is an example of a security policy:

```
{
"Version": "2012-10-17",
"Statement": [ {
"Action": [
"ec2:AllocateAddress",
"ec2:AssignPrivateIpAddresses",
"ec2:Associate*",
"ec2:Attach*",
"ec2:AuthorizeSecurityGroup*",
"ec2:Copy*",
"ec2:Create*",
"ec2:DeleteInternetGateway",
"ec2:DeleteNetworkAcl",
"ec2:DeleteNetworkAclEntry",
"ec2:DeleteNetworkInterface",
"ec2:DeletePlacementGroup",
"ec2:DeleteRoute",
"ec2:DeleteRouteTable",
"ec2:DeleteSecurityGroup",
"ec2:DeleteSnapshot",
"ec2:DeleteSubnet",
"ec2:DeleteTags",
"ec2:DeleteVolume",
"ec2:DeleteVpc",
"ec2:DeleteKeypair",
"ec2:Describe*",
"ec2:Detach*",
"ec2:DisassociateAddress",
"ec2:DisassociateRouteTable",
"ec2:EnableVolumeIO",
"ec2:GetConsoleOutput",
"ec2:ModifyImageAttribute",
"ec2:ModifyInstanceAttribute",
"ec2:ModifyNetworkInterfaceAttribute",
"ec2:ModifySnapshotAttribute",
"ec2:ModifyVolumeAttribute",
"ec2:ModifyVpcAttribute",
"ec2:MonitorInstances",
"ec2:ReleaseAddress",
"ec2:ReplaceNetworkAclAssociation",
"ec2:ReplaceNetworkAclEntry",
"ec2:ReplaceRoute",
"ec2:ReplaceRouteTableAssociation",
"ec2:ReportInstanceStatus",
"ec2:ResetImageAttribute",
"ec2:ResetInstanceAttribute",
```

Copyright © 2011 - 2017 EnterpriseDB Corporation. All rights reserved.

```
"ec2:ResetNetworkInterfaceAttribute",
"ec2:ResetSnapshotAttribute",
"ec2:RevokeSecurityGroup*",
"ec2:RunInstances",
"ec2:StartInstances",
"ec2:UnassignPrivateIpAddresses",
"ec2:UnmonitorInstances"
],
"Resource": "*",
"Effect": "Allow",
"Sid": "Stmt1407961327680"
}, {
"Action": [
"iam:PassRole"
],
"Resource": "*",
"Effect": "Allow",
"Sid": "Stmt1407961362664"
}, {
"Action": [
"s3:CreateBucket",
"s3:Get*",
"s3:List*"
],
"Resource": "*",
"Effect": "Allow",
"Sid": "Stmt1407961630932"
}, {
"Action": [
"s3:Put*",
"s3:Get*",
"s3:DeleteObject*"
],
"Resource": "arn:aws:s3:::*/wal_005*",
"Effect": "Allow",
"Sid": "Stmt1407961734627"
}, {
"Condition": {
"StringEquals": {
"ec2:ResourceTag/CreatedBy": "EnterpriseDB"
}
},
"Action": [
"ec2:RebootInstances",
"ec2:StopInstances",
"ec2:TerminateInstances"
],
"Resource": "*",
"Effect": "Allow",
"Sid": "Stmt1407961927870"
}
```

Copyright © 2011 - 2017 EnterpriseDB Corporation. All rights reserved.

```
]
}
```

Copyright © 2011 - 2017 EnterpriseDB Corporation. All rights reserved.

10.4 Reference – AWS User Trust Policy

When you define an Amazon role, you are required to provide a security policy. The following text is an example of a trust policy:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "",
      "Effect": "Allow",
      "Principal": {
        "Service": "ec2.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    },
    {
      "Sid": "",
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam:: your_account_number:root"
      },
      "Action": "sts:AssumeRole",
      "Condition": {
        "StringEquals": {
          "sts:ExternalId": "EDB-PPCD-CONSOLE"
        }
      }
    }
  ]
}
```

Copyright © 2011 - 2017 EnterpriseDB Corporation. All rights reserved.