EDB™ Ark

Administrative User's Guide

Version 2.3

January 26, 2018

EDB Ark Administrative User's Guide, Version 2.3
by EnterpriseDB® Corporation

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

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 2.1 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 EDB Ark -** Section 5 walks you through how to secure an EDB Ark cluster and opening a port for SSH connections.

- **Console Management** - Section 6 describes how to control the Ark console manager and customize the user console.

- **Recovering from a Console Failure -** Section 7 describes how to recover from a console failure.

- **Notifications –** Section 8 describes the user notifications that will keep you informed about any changes to your EDB Ark environment.

- **Resources** – Section 9 provides a list of EnterpriseDB resources that are available if you have unanswered questions.

- **References** – Section 10 provides reference information.

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

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

http://www.enterprisedb.com/documentation

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

1.1 What's New

The following features have been added to EDB Ark for release 2.3:

- Ark provisions EDB Postgres Advanced Server or PostgreSQL 10 clusters.
- Ark supports use of Postgres Authentication by the console's backing server; for more information, see Section 2.2.
- Ark engine definitions automatically provision the PEM agent; for more information, see Section 4.1.3.
- OpenStack Administrators can selectively allow access to the Ark administrative console to non-administrative OpenStack users. For more information, see Section 3.2.6.
- Ark supports statically provisioned server/engine definitions. A statically provisioned server is a pre-installed image that contains the software required to create a database cluster. For more information, see Section 4.1.2.
- You can use the `Edit Console Properties` dialog to manage the password of the backing database. For more information, see Section 6.3.
- SSL is now the default connection method for connections to the load balancer and the database on all nodes of a cluster.

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`".

1.3 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, Newton, and Ocata

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

- RHEL 7.x
- CentOS 7.x and 6.x

For information about the database engines supported by the Ark console, see Section 4.1.3.

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.

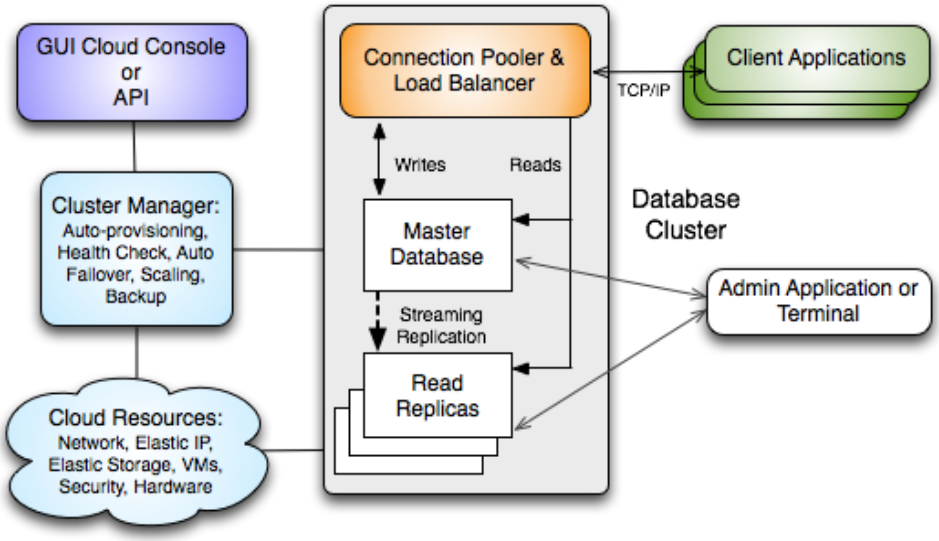

Figure 2.1 - An overview of the EDB Ark architecture.

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.

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.

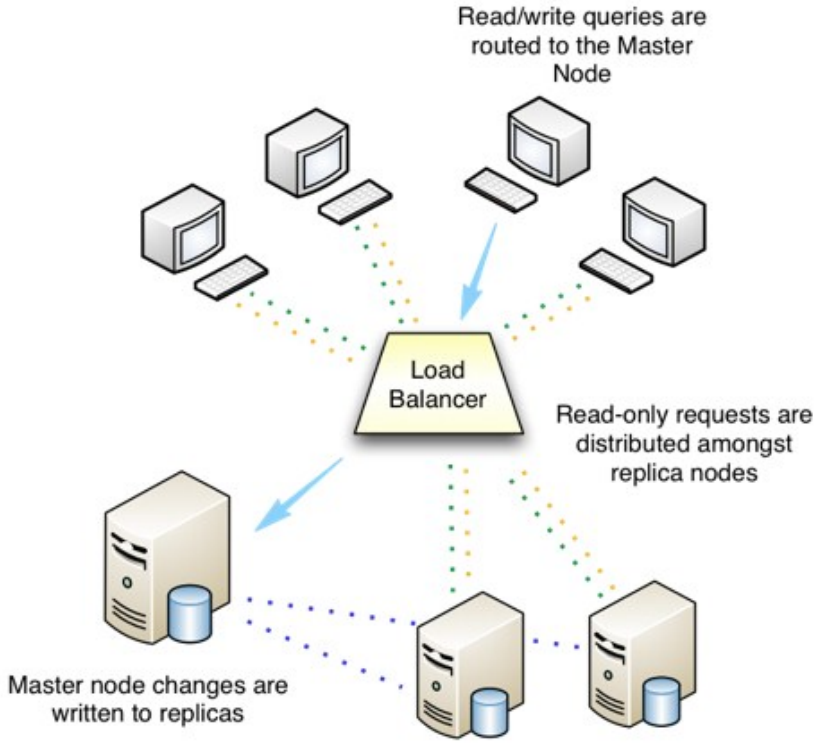

Figure 2.2 - EDB Ark performs automatic load balancing.

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.

EDB Ark makes it easy to scale a database cluster:

- To increase read performance, you can add read replicas to the cluster (manually or automatically).

- To handle expanding data requirements you can increase the amount of storage available (manually or automatically).

- To increase the RAM or CPU processing power of the cluster's underlying virtual machine, you can manually scale a cluster into a more appropriate server class.

2.1.1 Using EDB Ark on an OpenStack Host

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

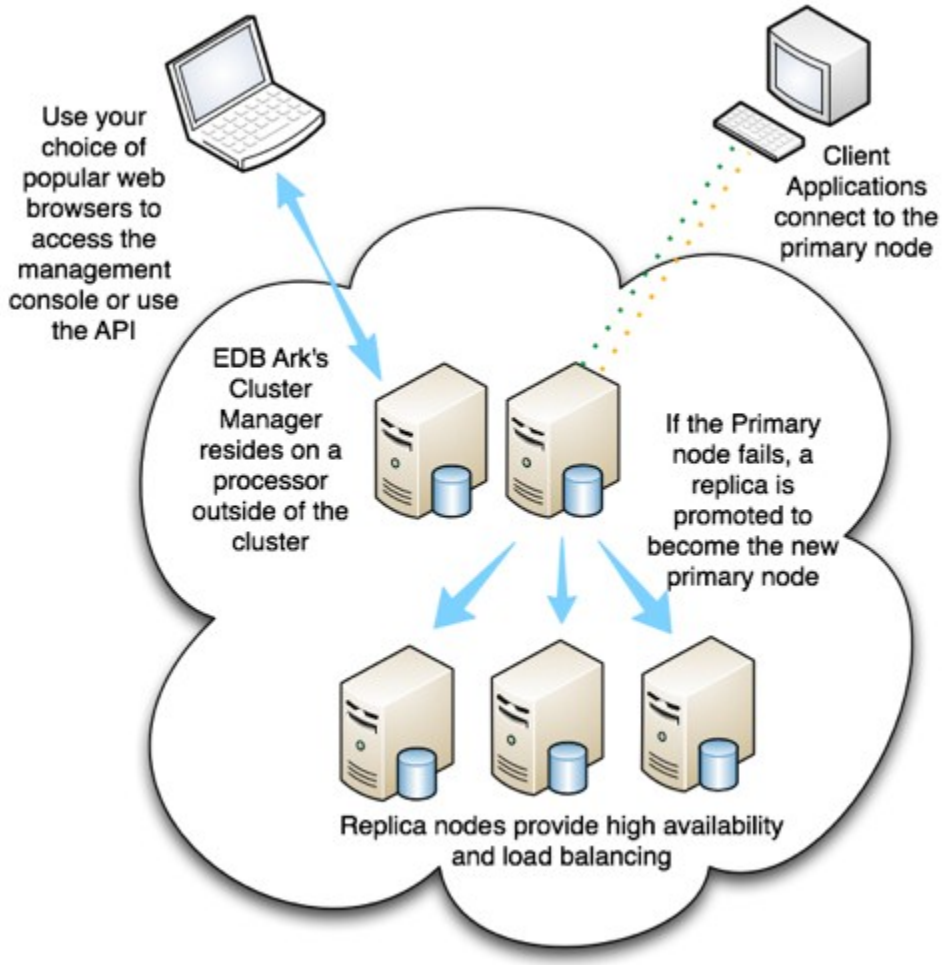

Figure 2.3 - Using EDB Ark in a Cloud.

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

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

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

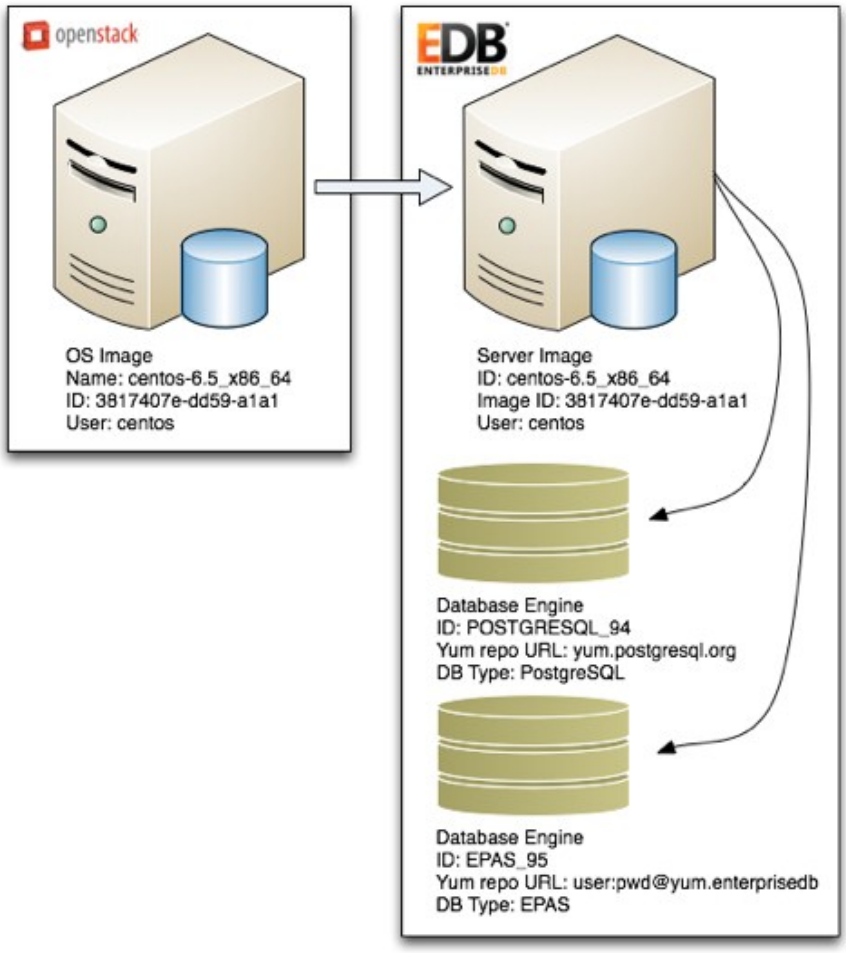

Figure 2.4 – Using an OpenStack image with EDB Ark.

The Administrator can use the same server image to create multiple database engine definitions. (see Figure 2.4). For example, you can create database engines for both `PostgreSQL 9.4` and `EDB Postgres Advanced Server 9.5` that are both provisioned on the same underlying server image `CentOS 6.5 x86_64`.

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

2.1.2 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.

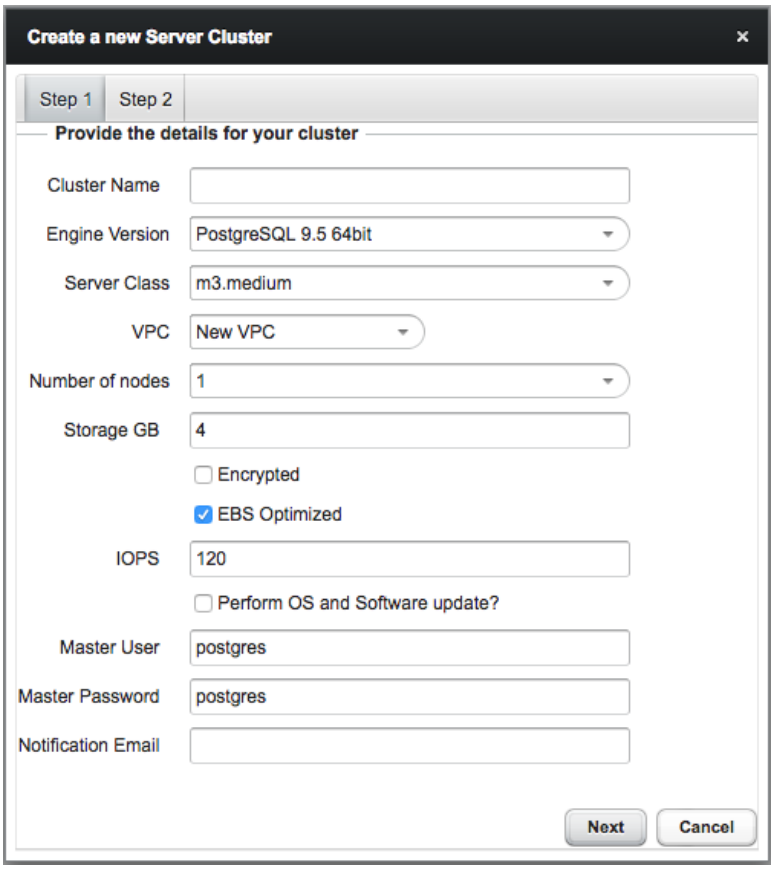

Figure 2.5 - Creating a new Ark cluster.

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.

2.2 Ark Authentication Models

When deploying the console, you can specify the type of authentication used by the Ark console. Authentication can be native password (provided by the service provider), or performed by the PostgreSQL backing database that resides on the host of the Ark console.

Using Native Password Authentication

When using native password authentication, an Administrative user must:

- On Amazon AWS: use the `User Administration` section of the Ark `Admin` tab to register Ark users.

- On OpenStack and Azure: use the OpenStack or Azure console to create user accounts and manage user access.

Using PostgreSQL Authentication

Ark supports using the following PostgreSQL authentication types:

- password
- LDAP
- RADIUS
- PAM
- BSD

For information about configuring authentication on a Postgres server, please consult the Postgres Core documentation, available at the EnterpriseDB website at:

https://www.enterprisedb.com/docs/en/10/pg/client-authentication.html

If you choose to use PostgreSQL authentication when deploying the Ark console, an Administrative user must:

- On Amazon AWS: add each user to the Ark backing database, and then use the `User Administration` section of the Ark `Admin` tab to register Ark users.

  Please note: On an Amazon host, the user name and associated password specified in the Ark backing database must match the credentials specified when registering the user in the Ark console.

- On Azure and OpenStack: add each user to the Ark backing database. Registration will be complete when the user logs in to the Ark console.

You can use the psql client to add a user to the `postgres` database. To use the psql client, SSH to the host of the Ark console; navigate into the `bin` directory, and 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 `CREATE ROLE` command to add a user to the database:

```
ADD USER user_name WITH PASSWORD 'password';
```

Where:

user_name specifies the name of the Ark user.

password specifies the password associated with the user name.

For detailed information about using the psql client please see the Postgres core documentation, available at:

https://www.enterprisedb.com/docs/en/10/pg/app-psql.html

After the administrative user adds the end-user, the end-user will complete the registration process by navigating to the URL of the console, and logging in.

2.2.1 Using Provider Authentication on Amazon

If you use authentication provided by Amazon, an Ark Administrative user can use the Ark Administrator's console to add, modify, or delete user accounts.

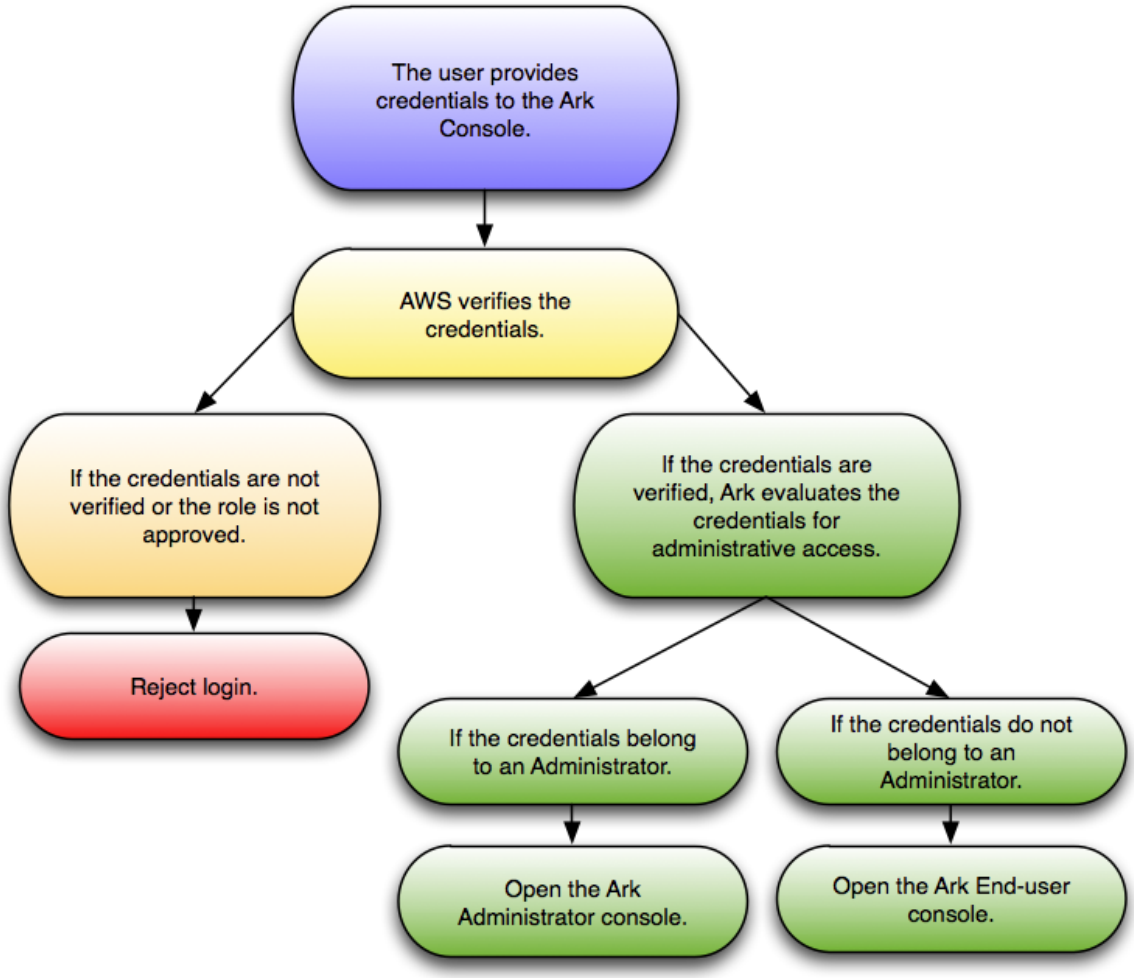

Figure 2.7 - Using provider authentication on Amazon.

When the user provides credentials to the Ark console, the credentials are passed to Amazon for verification. If the credentials are successfully verified, the role is evaluated to determine if the user should have access to the Administrator console or the End-user console.

2.2.2 Using PostgreSQL Authentication on AWS

When Postgres authentication is enabled, the first user to log in becomes the service user.

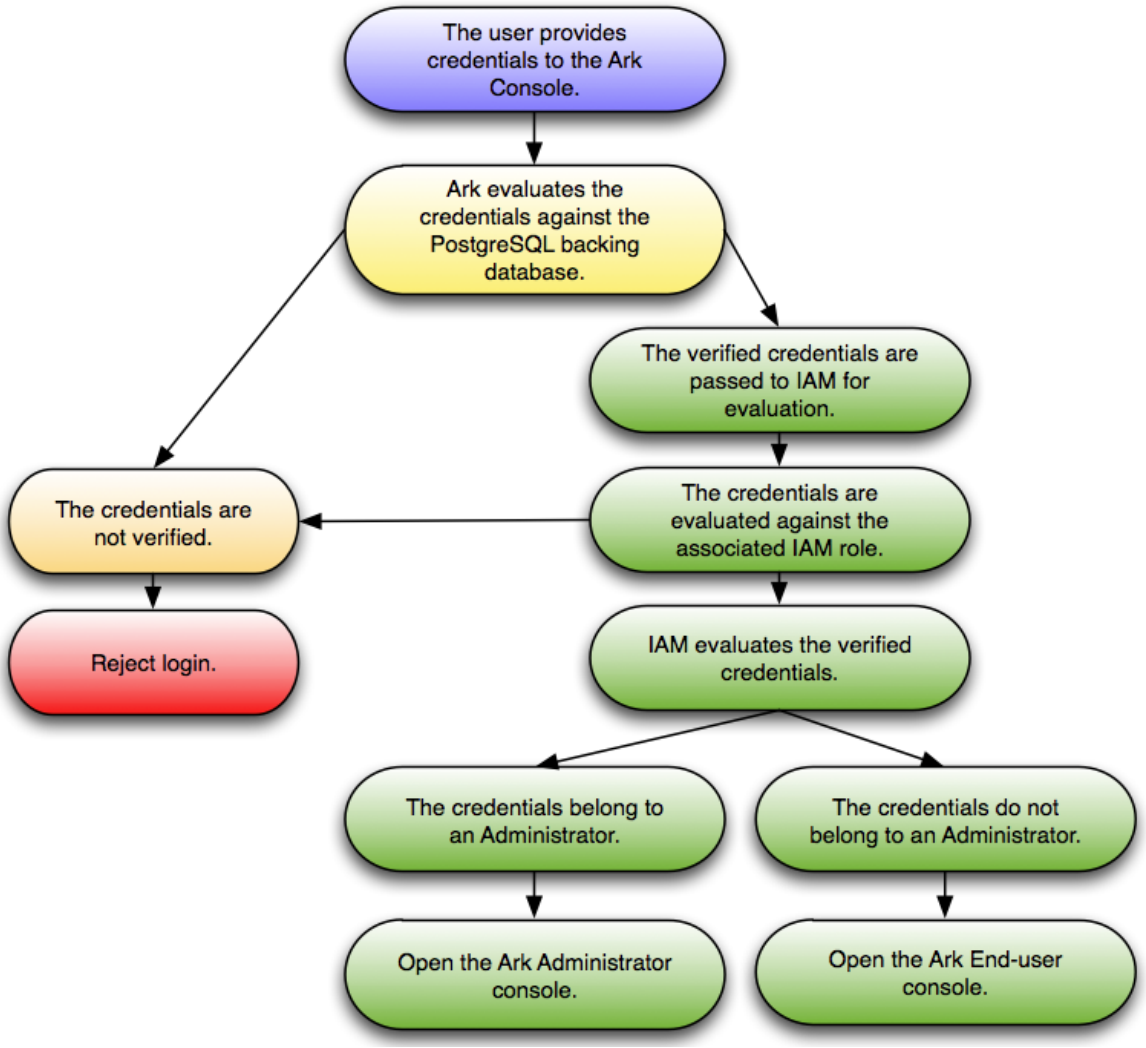

Figure 2.8 - Using Postgres authentication on AWS.

An Ark Administrative user must use a client application (such as psql or PEM) to add each user to the Ark backing database, and then use the `User Administration` table to register Ark users. The user name and associated password specified in the Ark backing database must match the credentials specified when registering the user in the Ark console. For more information, see Section 2.2.

If Ark successfully verifies the credentials, the credentials are passed to Amazon for evaluation to determine console access.

2.2.3 Using Provider Authentication on OpenStack or Azure

If you use native password authentication provided by OpenStack or Azure:

- On Azure: you must use the Azure Active Directory console to create and manage user accounts.
- On OpenStack: you must use the OpenStack administrator's console to create and manage user accounts.

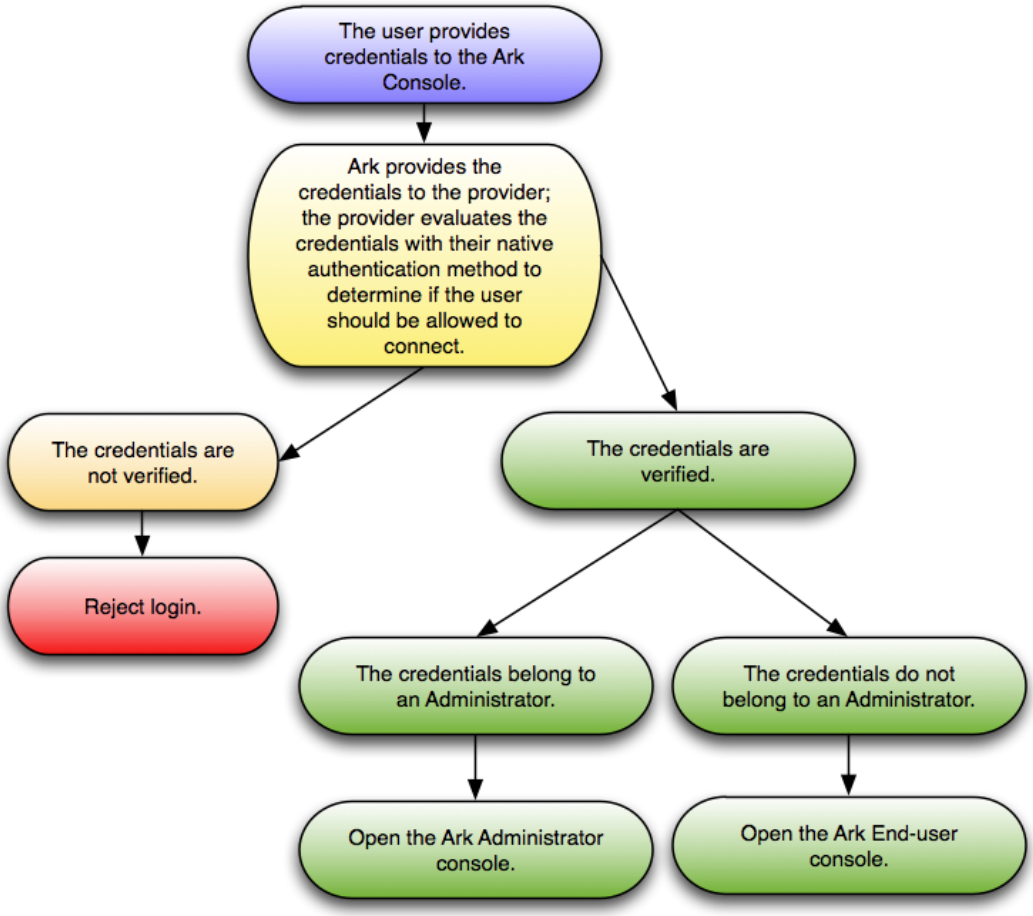

Figure 2.9 - Using provider authentication on OpenStack or Azure.

When the user provides credentials to the Ark console, the credentials are passed to the provider for verification. If the credentials are successfully verified, the role is evaluated to determine if the user should have access to the Administrator console or the End-user console.

2.2.4 Using PostgreSQL Authentication on OpenStack or Azure

When Postgres authentication is enabled on OpenStack or Azure, the first user to log in to the Ark console becomes the service user. An Administrator will be required to use either the PEM web interface or psql to add each successive user to the Ark backing database. User registration will be completed when the end user logs in to the Ark console.

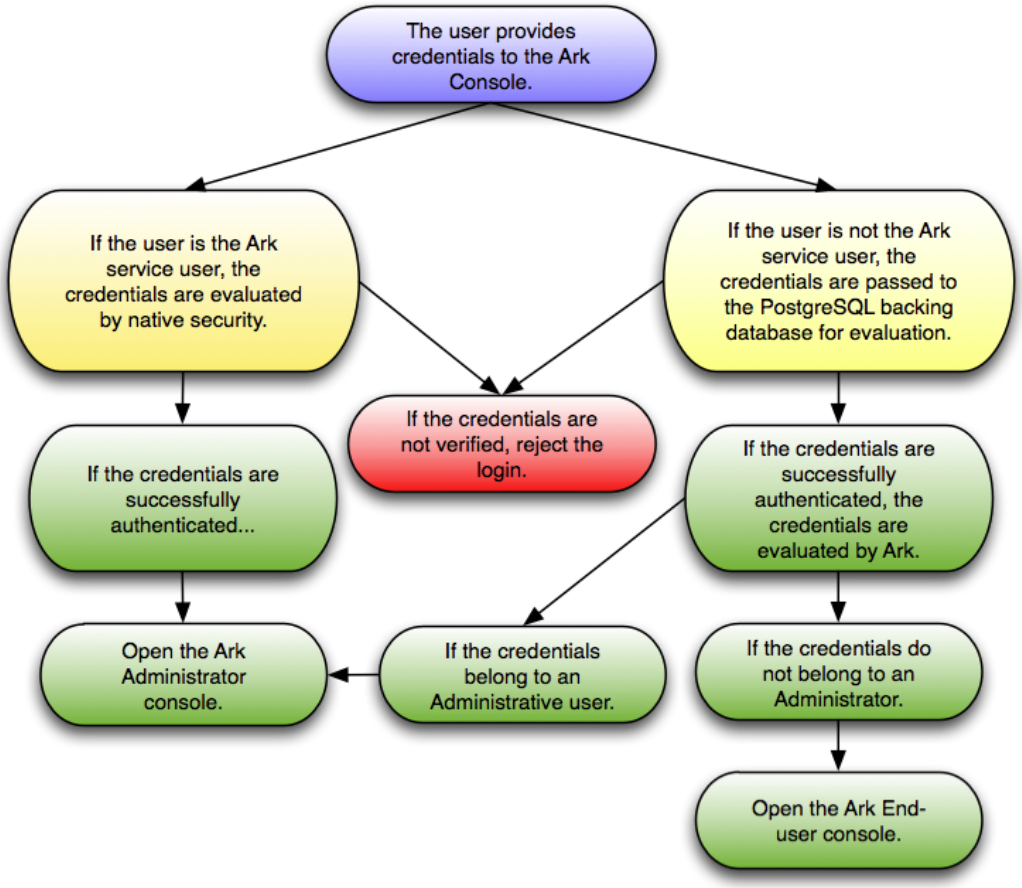

Figure 2.10 - Using Postgres authentication on OpenStack or Azure.

The credentials of the Ark service user are verified by the provider; all other credentials are verified by the Postgres server on the Ark console host. If Ark successfully verifies the credentials, the credentials are then evaluated to determine console access.

2.2.5 Using the OpenStack Standalone Security Model with PostgreSQL Authentication

If you use the OpenStack Standalone Security with authentication provided by the backing PostgreSQL server, the first user to connect to the Ark console becomes the Ark service user. An Administrative user must use either the PEM web interface or psql to add each user to the Ark backing database.

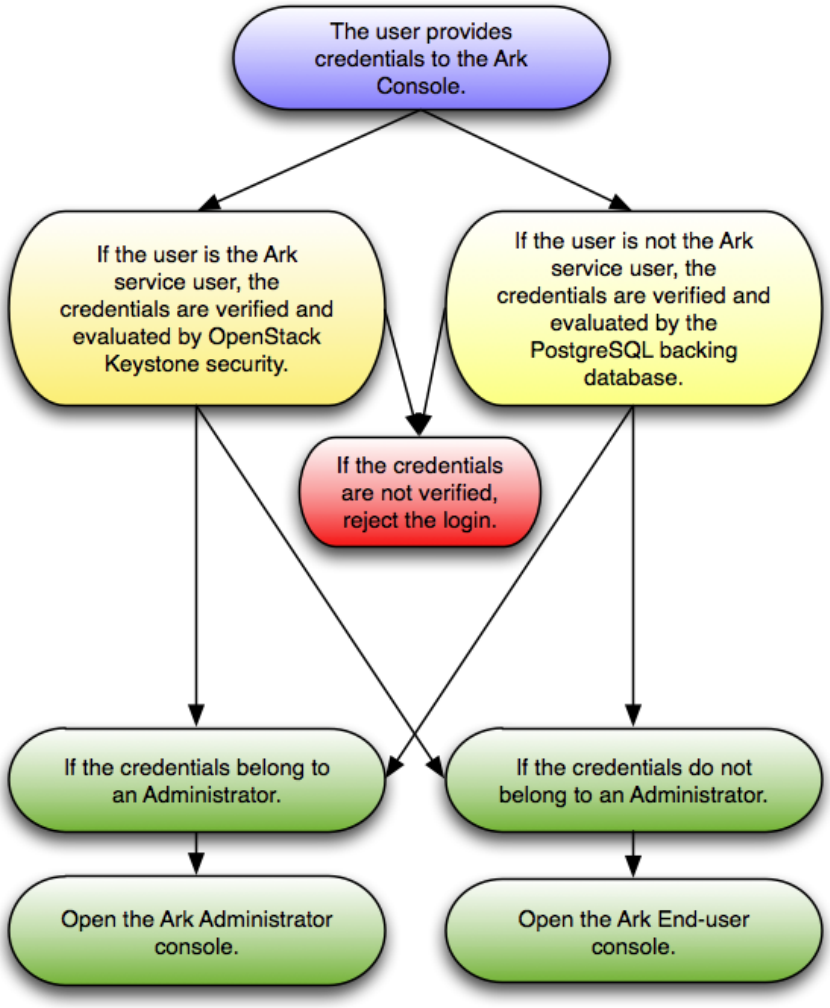

Figure 2.11 - Using Postgres authentication with OpenStack Standalone Security.

The credentials of the Ark service user are verified by OpenStack; all other credentials are verified by the Postgres server on the Ark console host. After the credentials are verified, they are evaluated to determine which console the user should be allowed to access.

3 Installing the EDB Ark Console

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.

After disabling pop-up blocker software for your console, follow the platform specific steps in the sections listed below to configure and deploy an Ark console:

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

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

- If your cluster uses an Azure host, see Section 3.3 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. Launch an Ark instance with an Amazon AWS Marketplace AMI. For more information, see Section 3.1.1.

2. Create an Amazon role and register an administrative user. For more information, see Section 3.1.2.

3. Configure the Ark console. For more information, see Section 3.1.3.

4. Create an Amazon role and register an Ark console user. For more information, see Section 3.1.4.

3.1.1 Launching the Ark Console Instance

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 launch an Amazon EC2 instance that contains a running copy of 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

Use the Amazon launch wizard to launch your instance, noting the requirements that follow on Step 3 and Step 6 of the wizard.

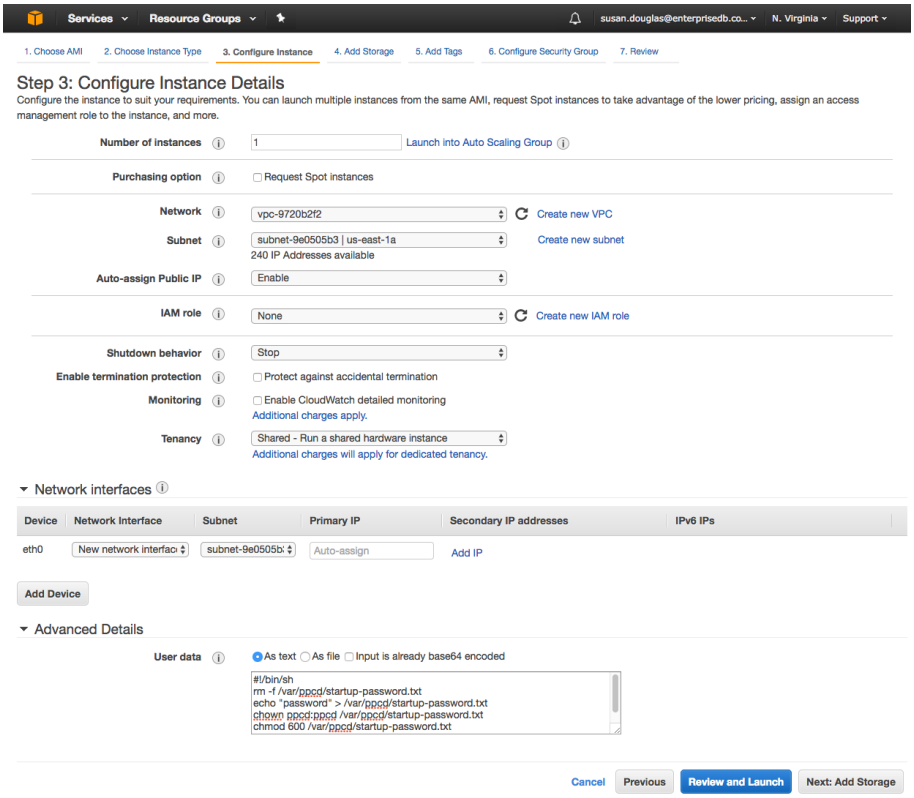

Figure 3.1 – Step 3 - Enabling the startup script.

When configuring your instance, you should include the following selections on the `Step 3: Configure Instance Details` dialog of the Amazon launch wizard (see Figure 3.1):

- Use the `Auto-assign Public IP` drop-down to specify `Enable` to automatically assign an IP address to the new instance.

- Use the `Advanced Details` section to provide the text of the script that will start the Ark console setup or recovery dialog.

  ```
  #!/bin/sh
  rm -f /var/ppcd/startup-password.txt
  echo "console_password" > /var/ppcd/startup-
  password.txt
  chown ppcd:ppcd /var/ppcd/startup-password.txt
  chmod 600 /var/ppcd/startup-password.txt
  ```

  When the user first connects to the AWS Ark console, they will be required to provide the *`console_password`* provided in the script.

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 (see Figure 3.2).

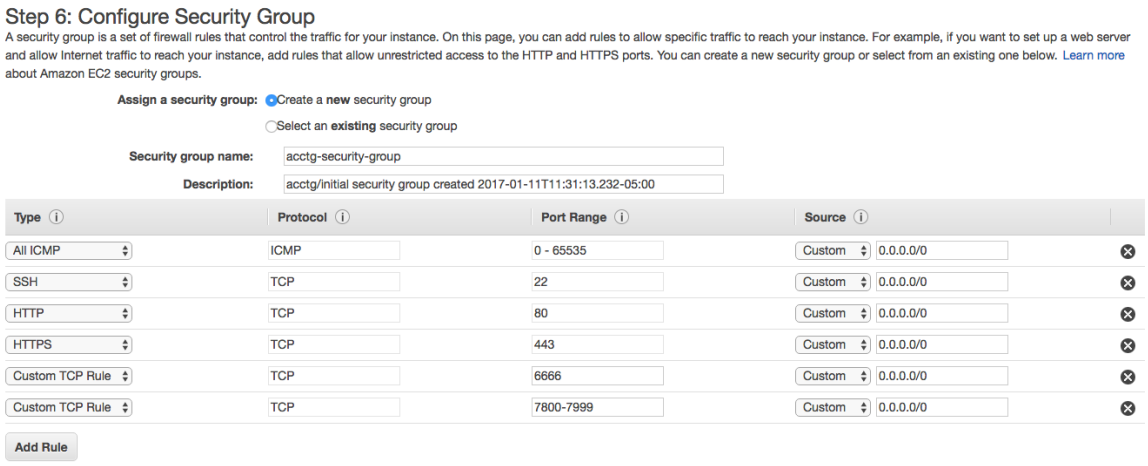

Figure 3.2 – Step 6 - Defining a Security Group.

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 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 the setup dialog with details about the AWS service user and the service role. Specify:

- the Amazon Role ARN (resource name) that will be used by the Ark service in the `Service Account Role ARN` field.
- the Amazon external ID that will be used by the Ark service user (`ppcd`) in the `Service Account External ID` field.
- the `AWS_ACCESS_KEY_ID` associated with the AWS role used for account administration in `AWS Access Key` field.
- the `AWS_SECRET_ACCESS_KEY` associated with the AWS role used for account administration in `AWS Secret Key` field.

3.1.2.1 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 3.3).

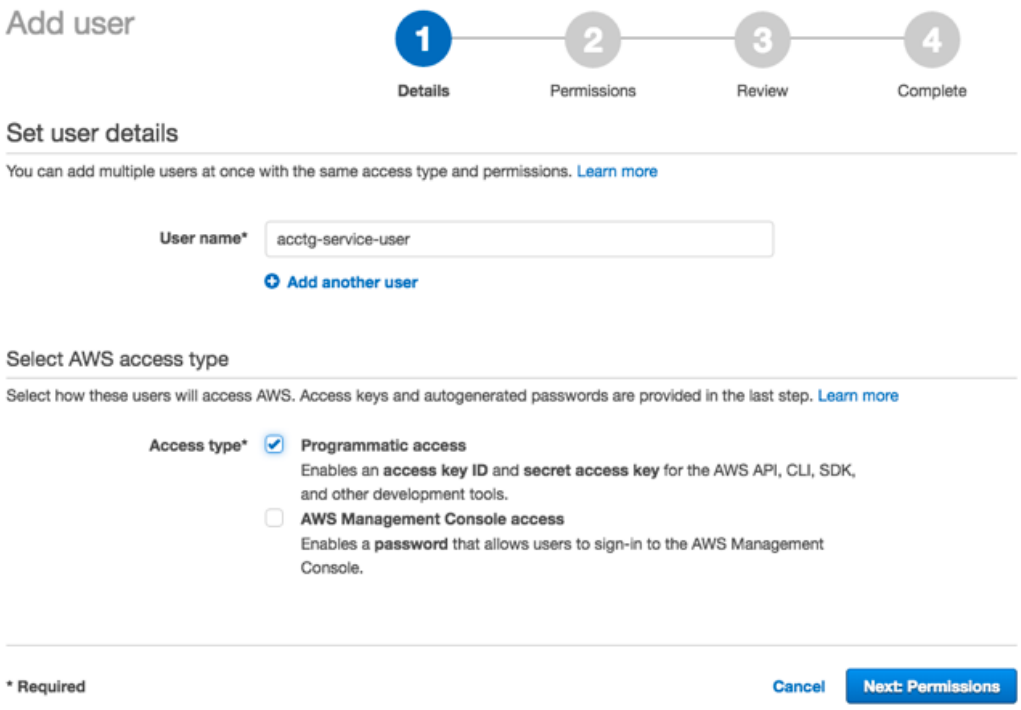

Figure 3.3 - The Add user dialog.

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 `Create Policy` button.

The browser will open another tab, allowing you to define a custom policy (see Figure 3.4).

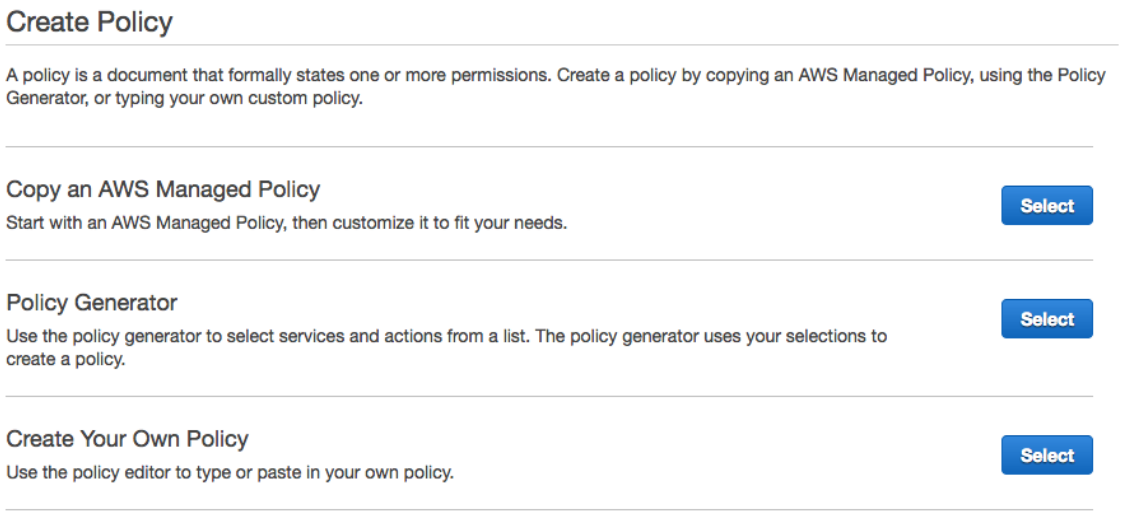

Figure 3.4 - The Create Policy dialog.

Click the `Select` button to the right of `Create Your Own Policy` to provide a security policy.

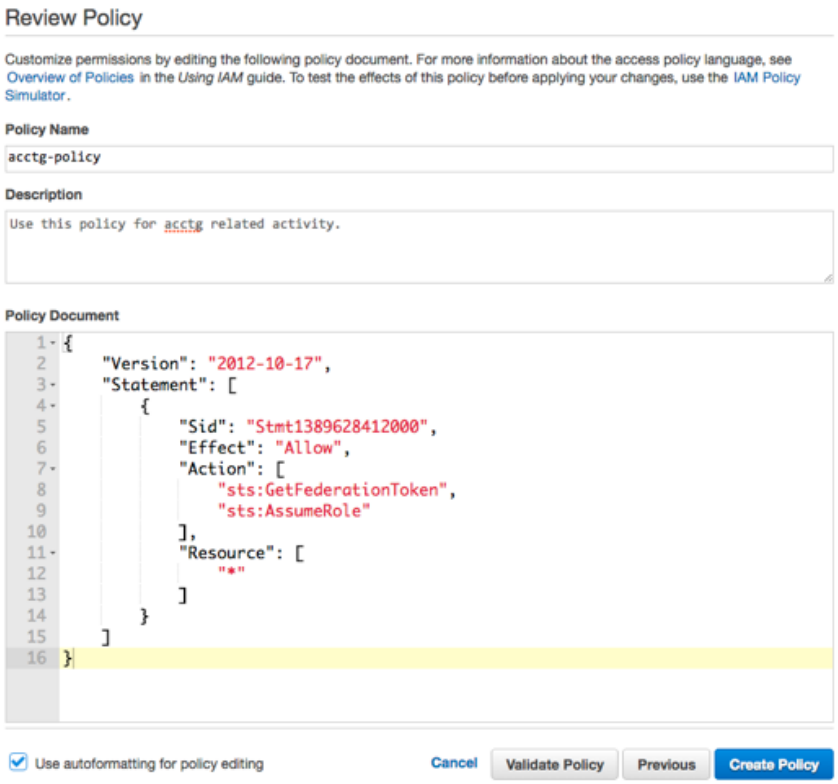

Figure 3.5 - The Review Policy dialog.

On the `Review Policy` dialog (see Figure 3.5):

- 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 Section 10.1.

Click `Create Policy` to continue.

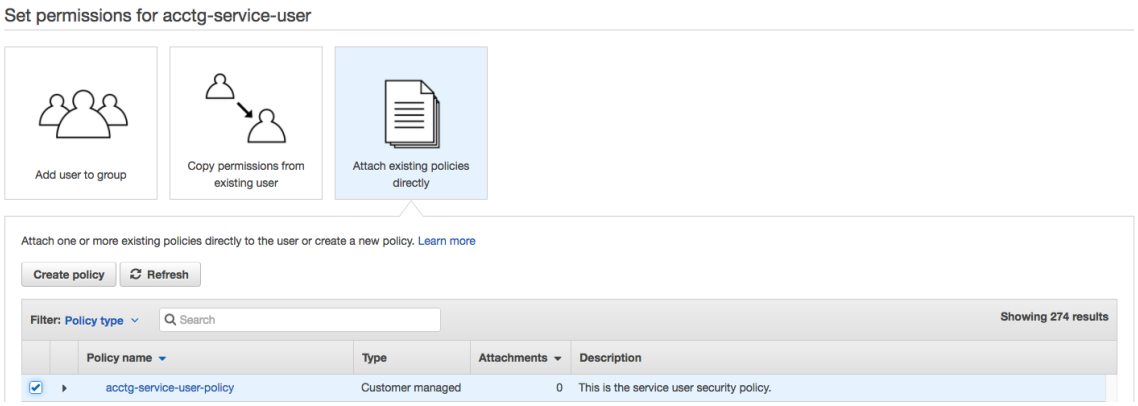

Figure 3.6 – Review the Add user dialog.

Then, return to the `Add user` tab, and click the `Refresh` button above the list of policies (see Figure 3.6). Select the new policy from the list, and click `Next: review`.

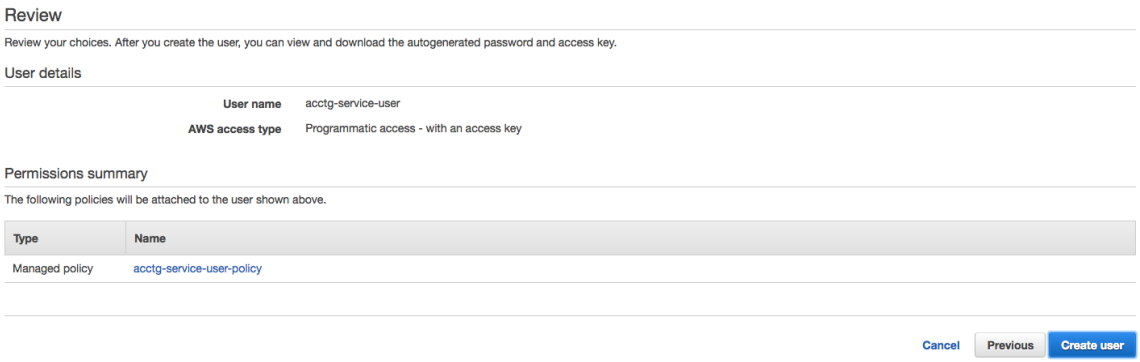

Figure 3.7 – Creating the user.

Review the details about the user account, and click the `Create user` button to create the user (see Figure 3.7).

The AWS console will confirm that the user has been added successfully. Click `Show` to display the `Secret access key` value (see Figure 3.8).

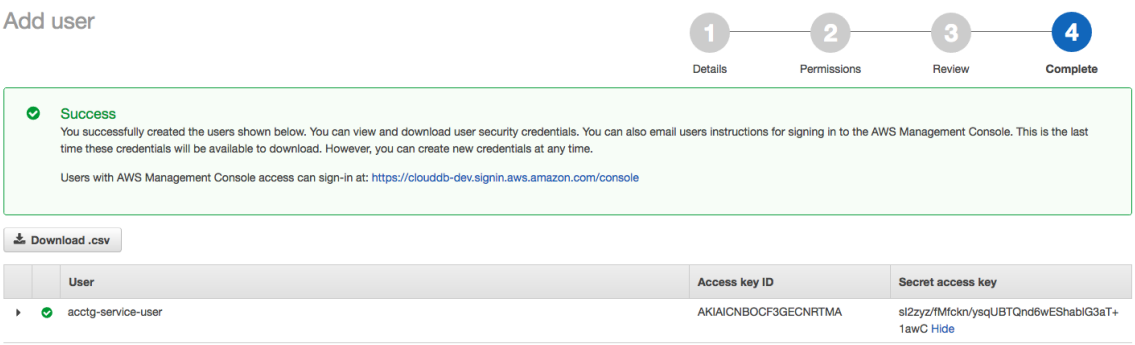

Figure 3.8 – Retrieving Access key information.

Copy the access key values displayed on the console; you must provide the values when configuring your Ark console:

- Provide the `Access key id` in the `AWS Access Key` field on the Ark console setup dialog.
- Provide the `Secret access key` in the `AWS Secret Key` field on the Ark console setup dialog.

3.1.2.2 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 3.9).

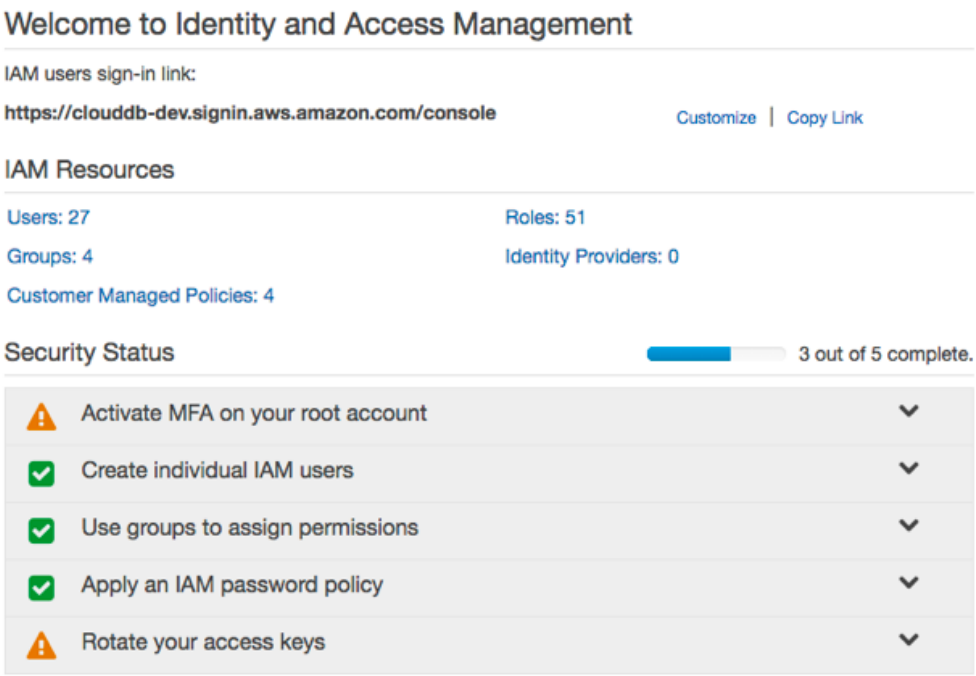

Figure 3.9 - The Amazon IAM Dashboard.

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

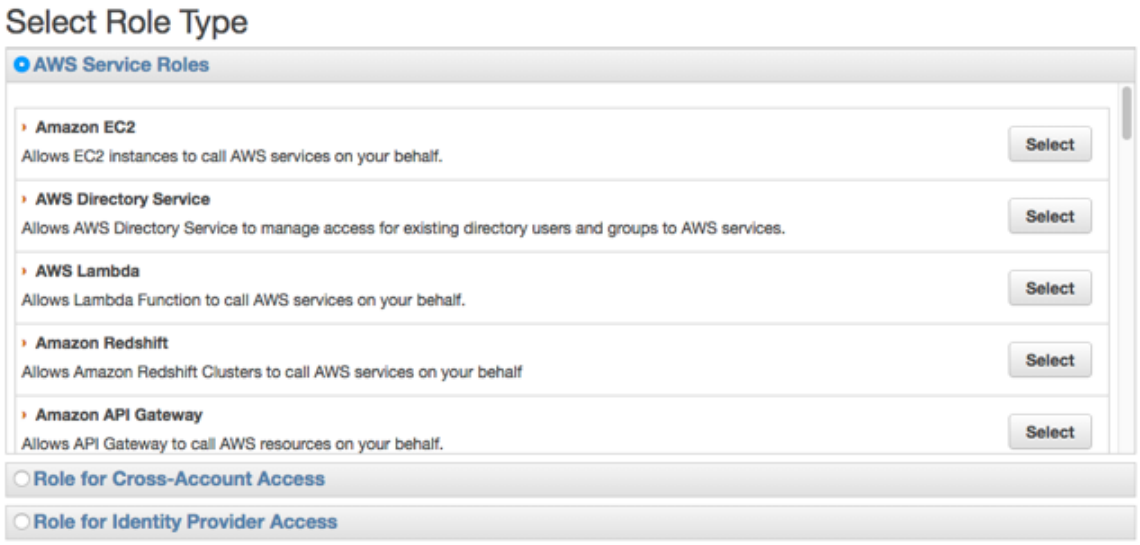

Figure 3.10 - Specify that the role allows EC2 instances to call AWS services.

Select the `AWS Service Roles` radio button (shown in Figure 3.10), and then the `Select` button to the right of `Amazon EC2` to continue to the `Attach Policy` dialog.

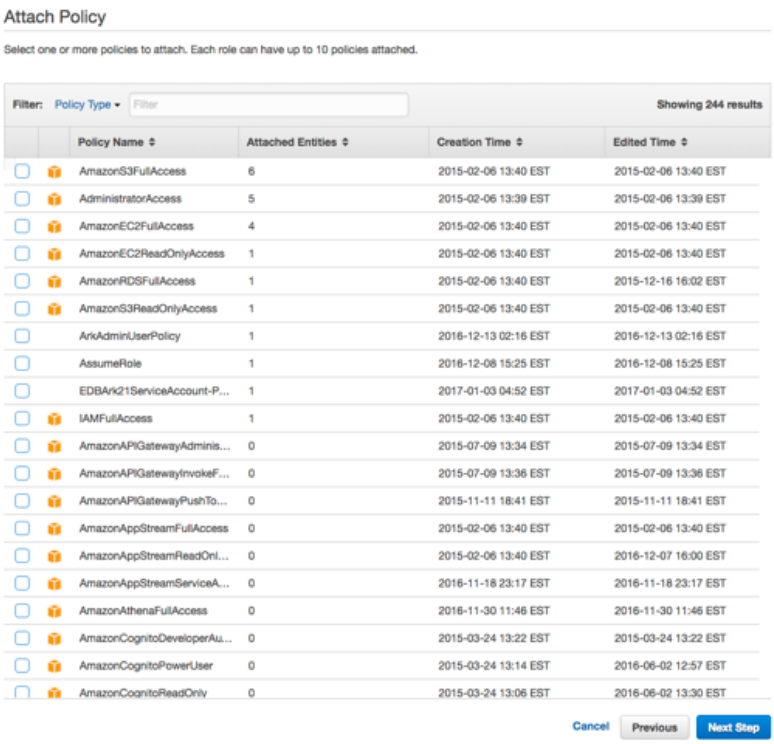

Figure 3.11 – The Attach Policy dialog.

When the `Attach Policy` dialog (shown in Figure 3.11) opens, do not select a policy; instead, click `Next Step` to continue to the `Set role name and review` dialog.

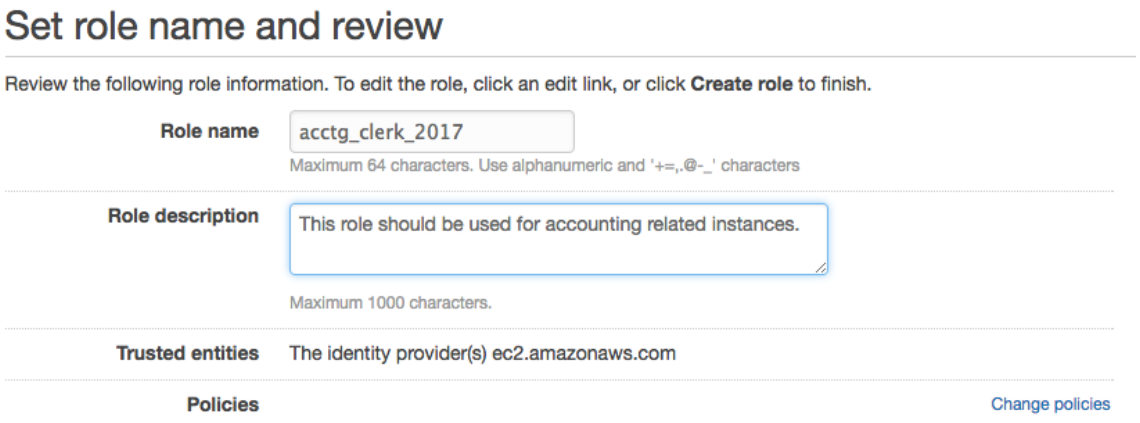

Figure 3.12 - Provide a role name.

When the `Create Role` dialog opens (shown in Figure 3.12), specify a name for the new role and click the `Create Role` button.

Figure 3.13 - 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 3.13). 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, navigate to the `Permissions` tab, expand the `Inline Policies` menu, and select `click here` to add a new policy (see Figure 3.14).

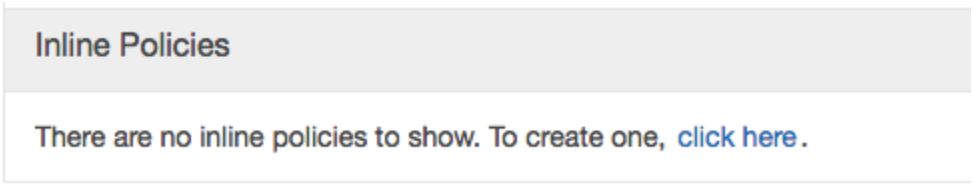

Figure 3.14 - The Inline Policies menu.

When the `Set Permissions` dialog opens, select the `Custom Policy` radio button, and then click the `Select` button (see Figure 3.15).

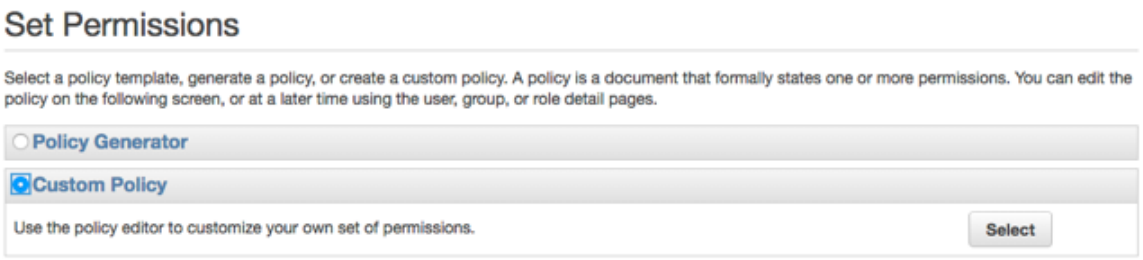

Figure 3.15 - Add a Custom Policy.

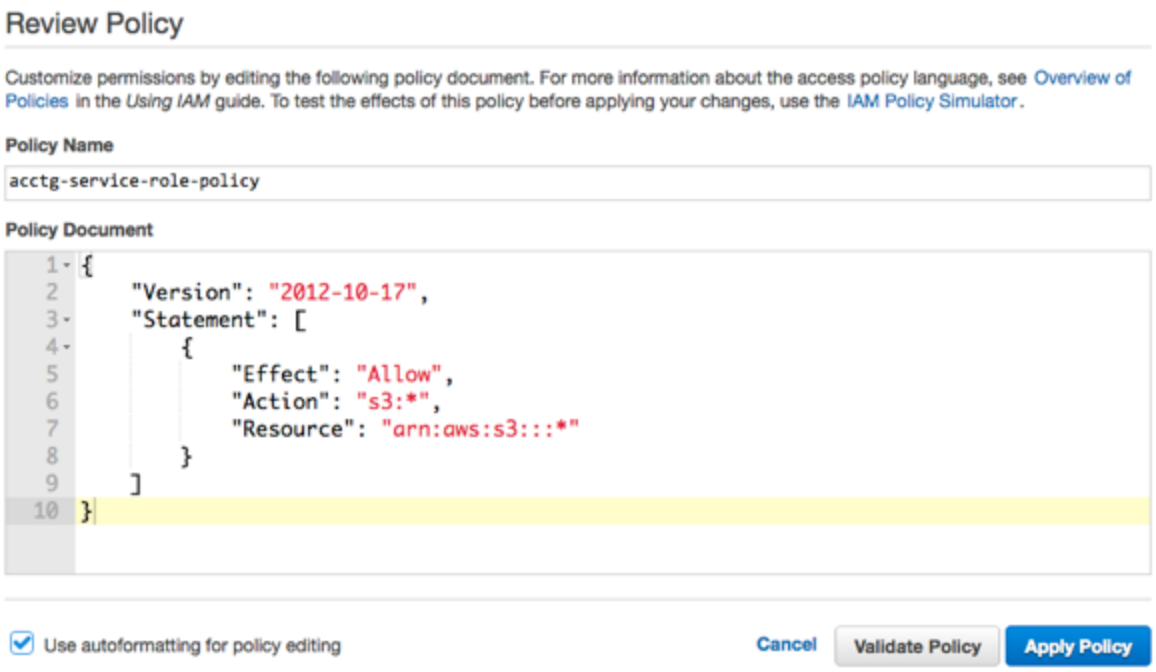

Figure 3.16 - Provide the policy name and contents.

Use the fields on the `Set Permissions` dialog (Figure 3.16) 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 3.17).

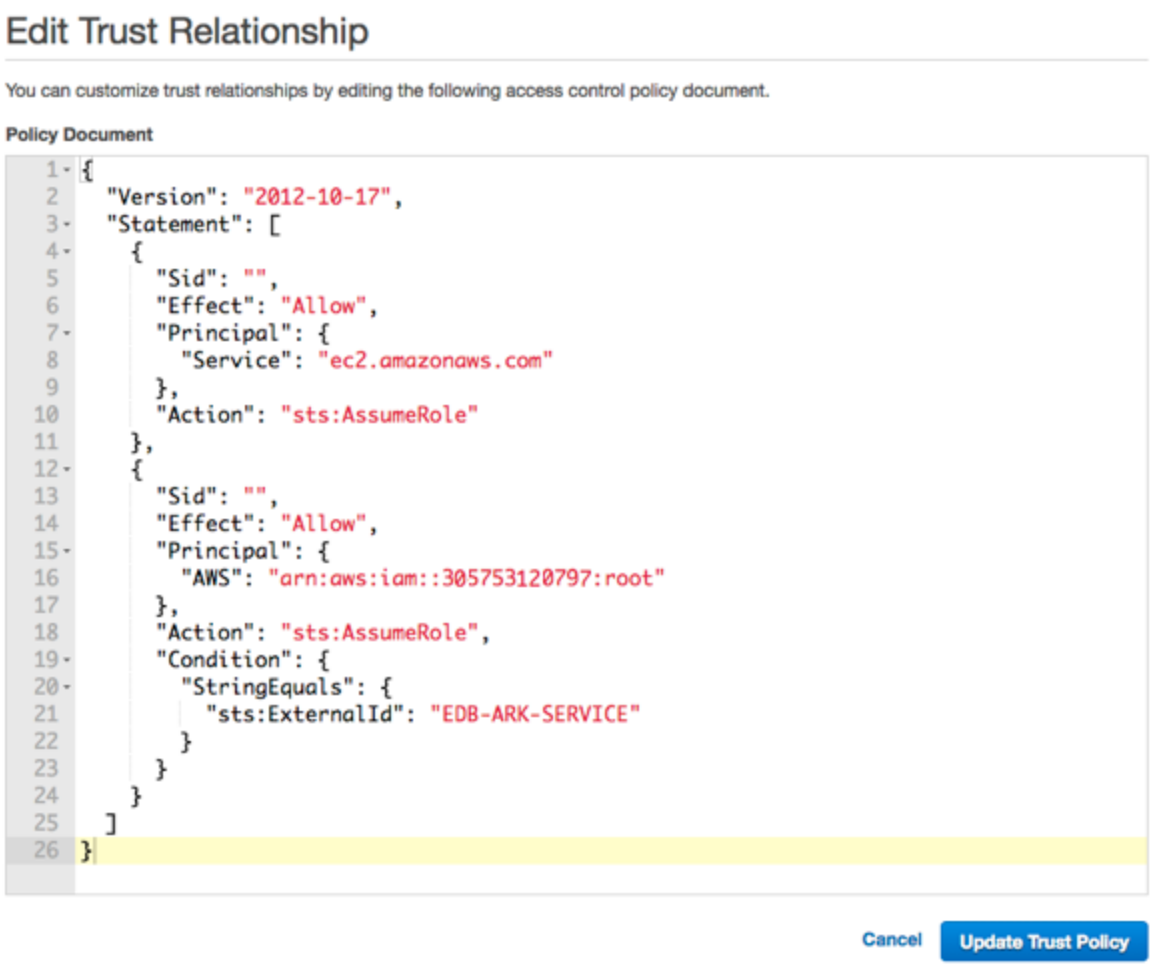

Figure 3.17 - The Policy Document.

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.

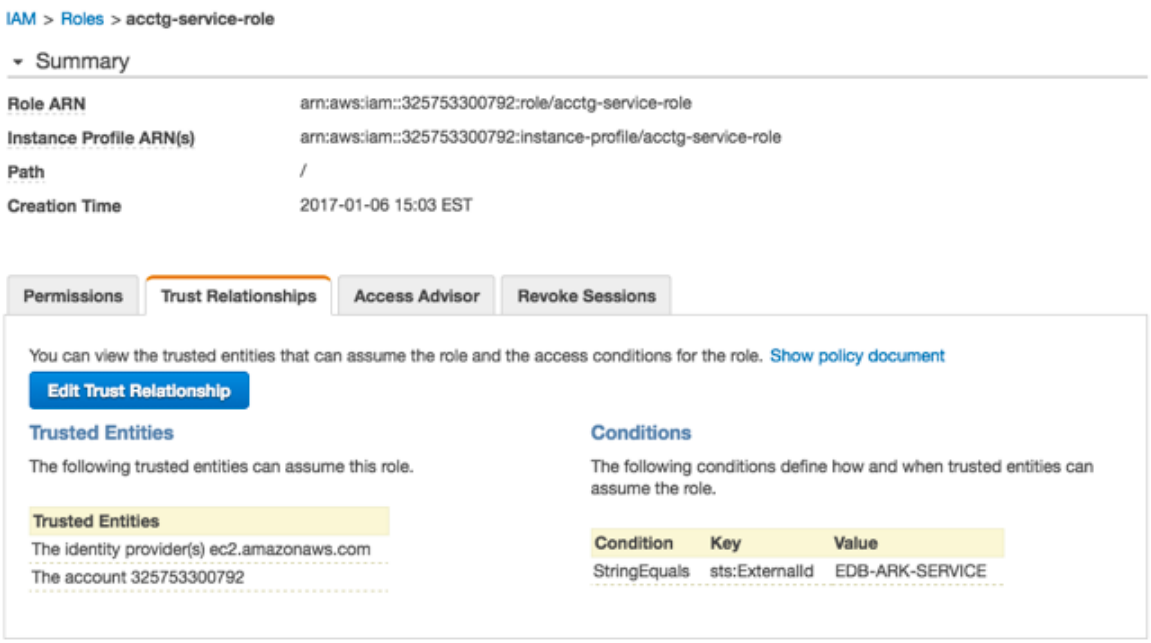

Figure 3.18 - The Summary tab of the Role detail panel.

The `Summary` dashboard (see Figure 3.18) will display values that you must provide when configuring your Ark console:

- The `Role ARN` associated with the service role must be provided in the `Service Account Role ARN` field.
- The external ID associated with the service role must be provided in the `Service Account External ID` field. 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.

3.1.3 Configuring the Console

After launching the instance, navigate to the public IP address of the cluster. When prompted, provide the password specified when launching the console in the `Password` field (see Figure 3.19), and click `Submit`.

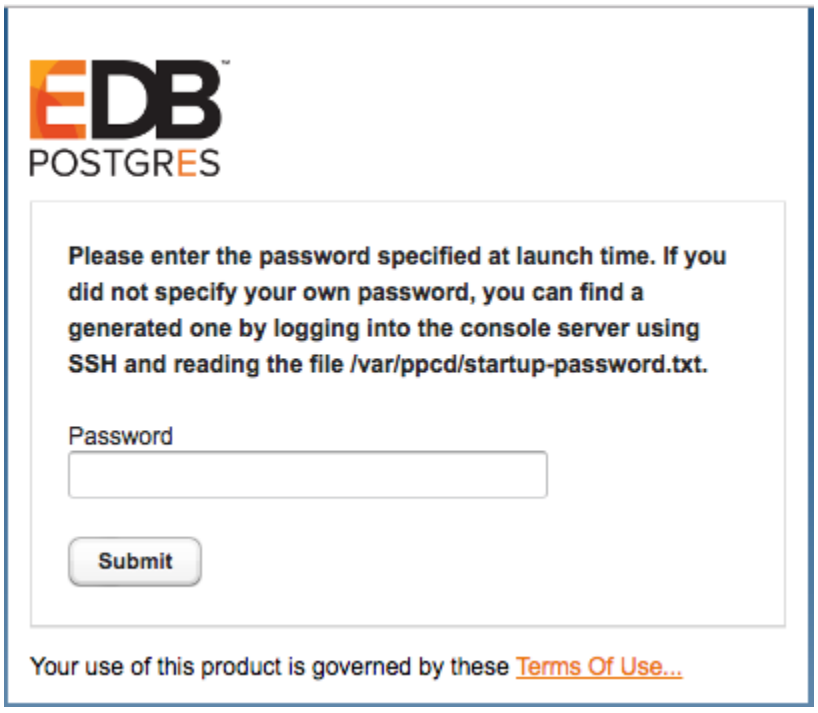

Figure 3.19 – Starting the setup dialog.

The installation properties dialog opens (as shown in Figure 3.20).

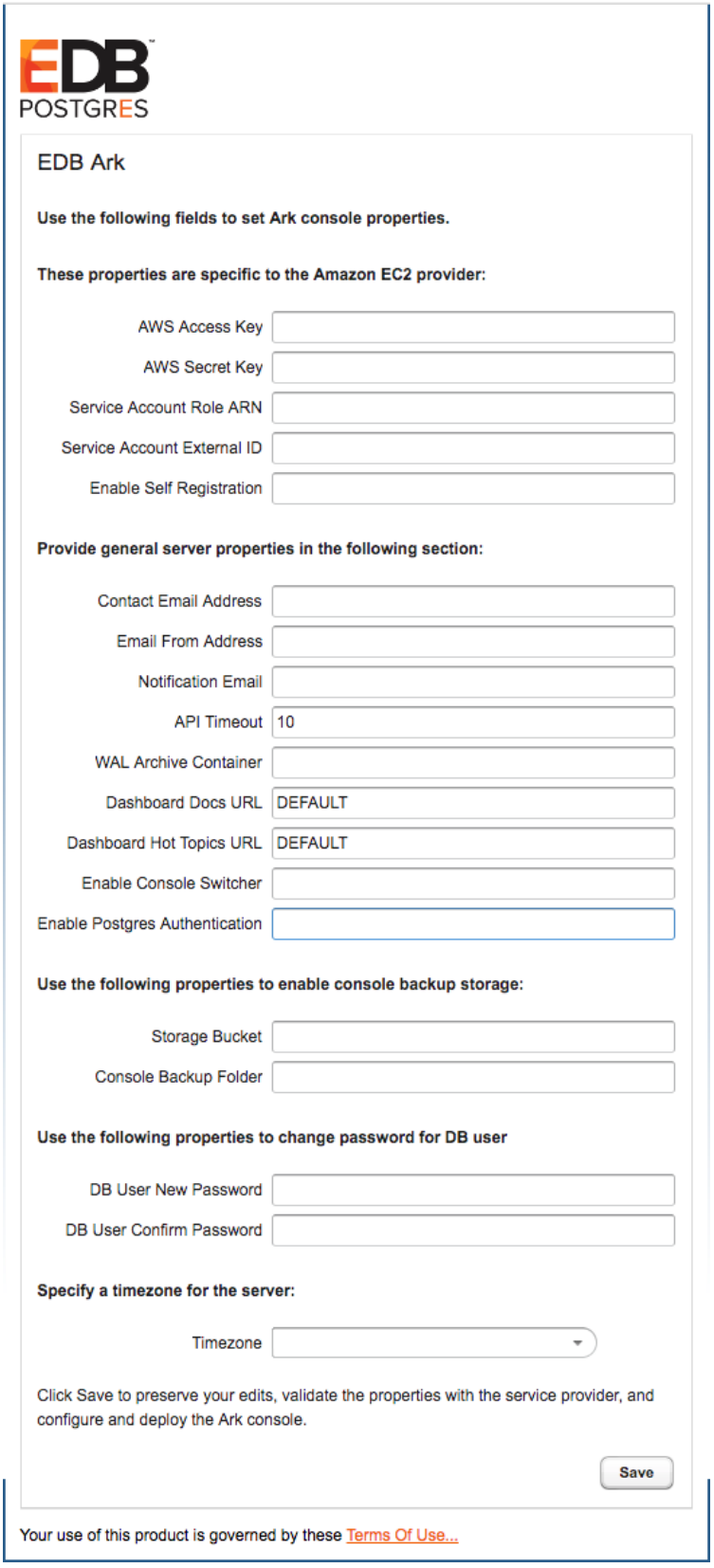

Figure 3.20 – The console setup dialog.

Complete the setup dialog, providing values that are specific to your Amazon account in the first section:

- Use the `AWS Access Key` field to specify the Amazon access key ID associated with the AWS role that will be used for account administration.

- Use the `AWS Secret Key` field to specify the Amazon secret key associated with the AWS role that will be used for account administration.

- Use the `Service Account Role ARN` field to specify the Amazon Role ARN (resource name) that should be used by the Ark service user (`ppcd`) when performing management functions on behalf of Ark.

- Use the `Service Account External ID` field to specify the Amazon external ID that should be used by the Ark service user (`ppcd`).

- Use the `Enable Self Registration` field to specify if the Ark console should allow self-registration for Ark users; specify `true` to allow self-registration, or `false` to disable self-registration.

Provide general server properties in the next section:

- Use the `Contact Email Address` field to specify the email address that will be included in the body of cluster status notification emails.

- Use the `Email From Address` field to specify the return email address used on cluster status notification emails.

- Use the `Notification Email` field to specify the email address to which email notifications about the status of the Ark console will be sent.

- Use the `API Timeout` field to specify the number of minutes that an authorization token will be valid for use with the API.

- Use the `WAL Archive Container` field to specify 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 field; once set, this property must not be changed.

    - The bucket name must be at least 3 and no more than 63 characters long.

    - The name can contain lowercase letters, numbers, and hyphens; the name must start with and end with a lowercase letter or number.

    - A series of one or more labels; adjacent labels are separated by a single period (.). A name may not be formatted as an IP address.

For more information, please visit:

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

- Use the `Dashboard Docs URL` field to specify the location 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 parameter to `DEFAULT` to display content (documentation) from EnterpriseDB; to display alternate content, provide the URL of the content. To display no content in the lower half of the `Dashboard` tab, leave the field blank.

- Use the `Dashboard Hot Topics URL` field to specify the location 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 parameter to `DEFAULT` to display content (alerts) from EnterpriseDB; to display alternate content, provide the URL of the content. Leave the field blank to omit content.

- Use the `Enable Console Switcher` field to indicate if the console should display console switcher functionality; for more information, see Section 4.1.1.

- Set `Enable Postgres Authentication` to `true` to instruct Ark to enforce the authentication method configured on the backing Postgres server. Supported authentication methods include password, LDAP, RADIUS, PAM, and BSD.

  If `false`, Ark will use the default authentication method (password).

Use the next section to specify your console backup storage preferences:

- Use the `Storage Bucket` field to specify the name of the bucket in which backups will be stored.

- Use the `Console Backup Folder` field to specify the name of the backup folder within the storage bucket.

Use fields in the next section to specify database password preferences for the database superuser (`postgres`) on the backing PostgreSQL database (`postgres`):

- Use the `DB User New Password` field to set the password for the `postgres` user on the console's backing database (`postgres`).

- Use the `DB User Confirm Password` field to set the password for the `postgres` user on the console's backing database (`postgres`).

Use the last field to specify a timezone for the server:

- Use the drop-down listbox in the `Timezone` field to select the timezone that will be displayed by the Ark console.

When you've completed the setup dialog, click the `Save` button to validate your changes.

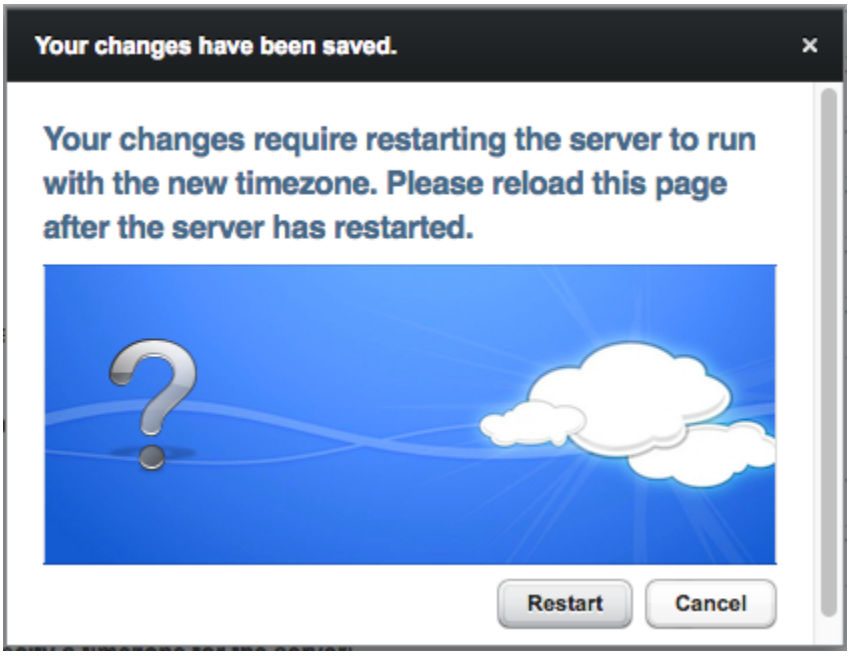

Figure 3.21 – Restart the server to start the Ark console.

When prompted, click the `Restart` button to restart the server and start the Ark console. Ark will confirm that the server is restarting (see Figure 3.22).

Figure 3.22 – The server restart message.

When the server has finished restarting, refresh your browser; the Ark console will prompt you to register a user (see Figure 3.23).

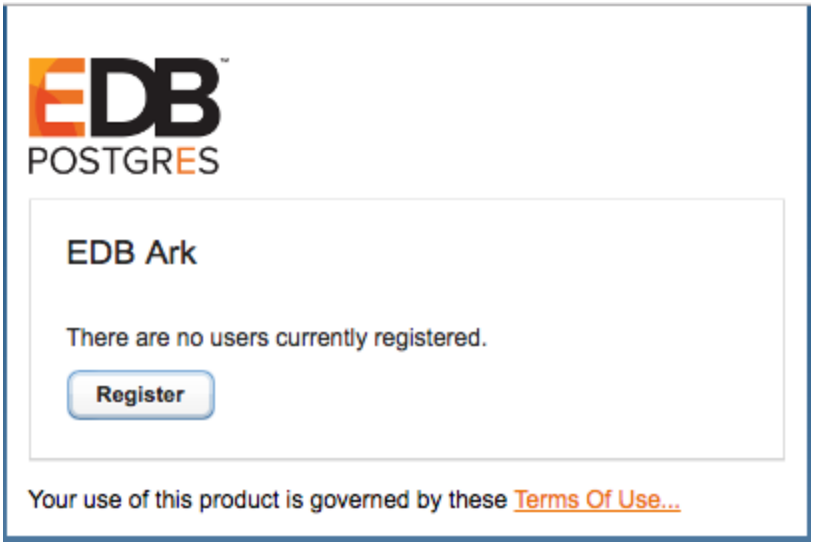

Figure 3.23 – Register an Ark user.

The first user registered will be the Ark service user, and will have access to the administrator's console.

3.1.4 Creating an Amazon Role and Registering an Ark Console User

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 3.24).

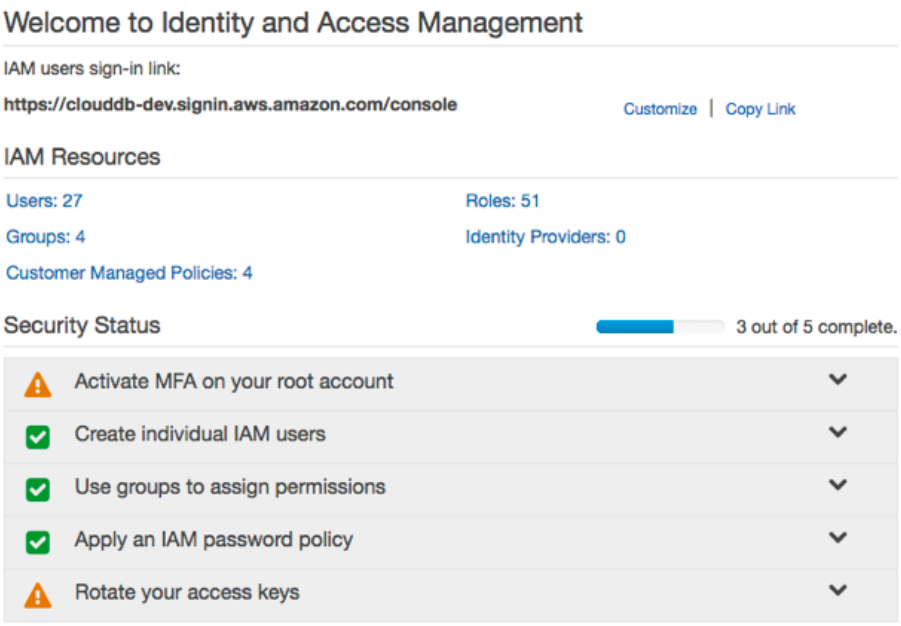

Figure 3.24 - The Amazon IAM Dashboard.

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

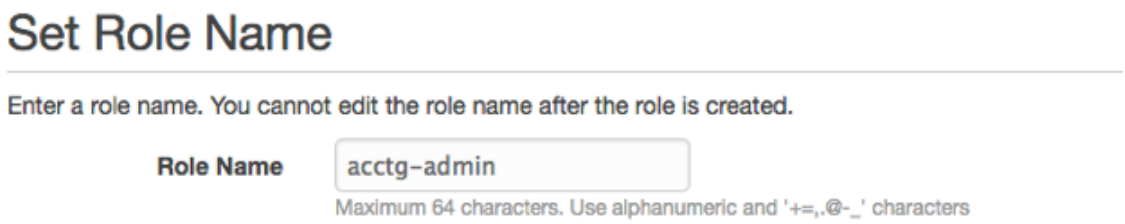

Figure 3.25 - Provide a role name.

When the `Set Role Name` dialog opens (shown in Figure 3.25), specify a name for the new role and click `Next Step` to select a role type.

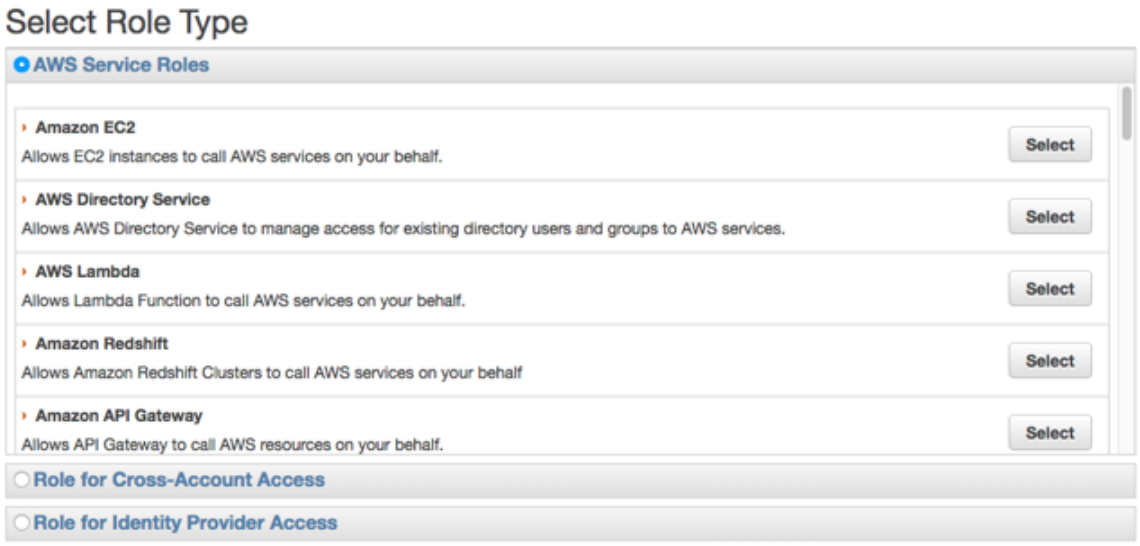

Figure 3.26 - Specify that the role allows EC2 instances to call AWS services.

On the `Select Role Type` dialog, select the `AWS Service Roles` radio button (shown in Figure 3.26), and then the `Select` button to the right of `Amazon EC2` to continue to the `Attach Policy` dialog.

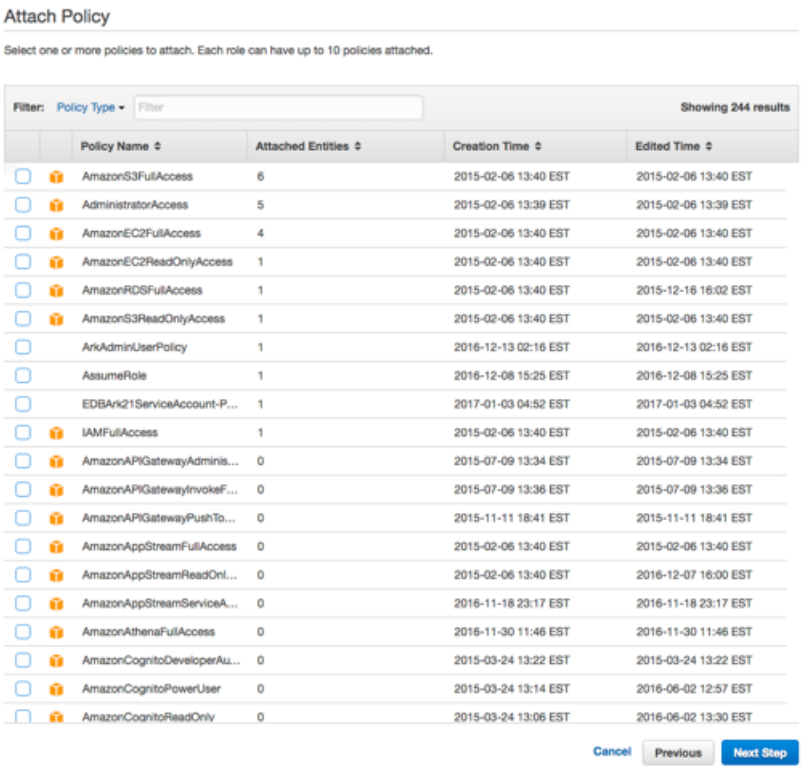

Figure 3.27 – The Attach Policy dialog.

When the `Attach Policy` dialog (shown in Figure 3.27) opens, do not specify a policy; instead, click `Next Step` to continue to the `Review` dialog.

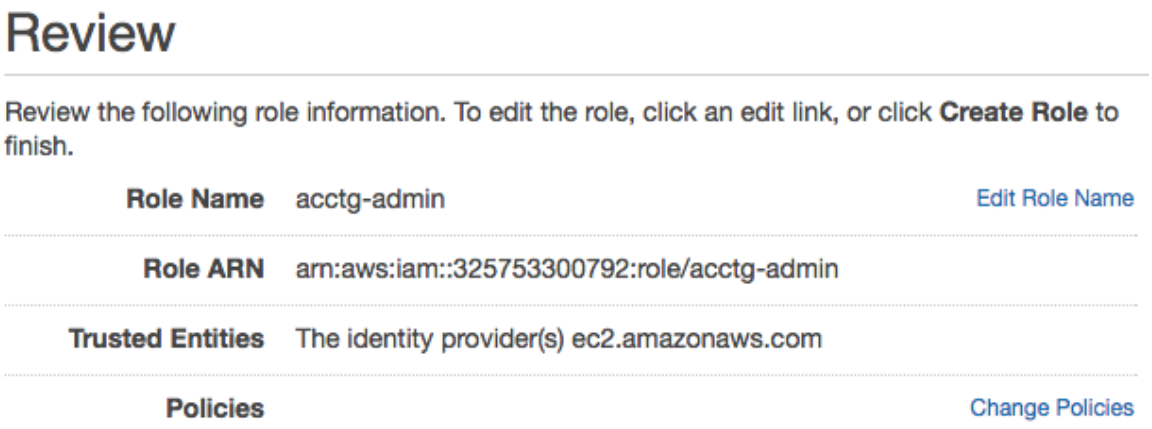

Figure 3.28 - Review the role information.

When the `Review` dialog opens (as shown in Figure 3.28), review the information displayed, and then click `Create Role` to instruct the AWS management console to create the described role.

Figure 3.29 - 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 3.29). 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, navigate to the `Permissions` tab, expand the `Inline Policies` menu, and select `click here` to add a new policy (see Figure 3.30).

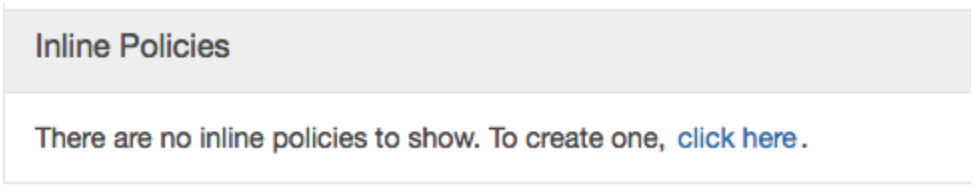

Figure 3.30 - The Inline Policies menu.

When the `Set Permissions` dialog opens, select the `Custom Policy` radio button, and then click the `Select` button (see Figure 3.31).

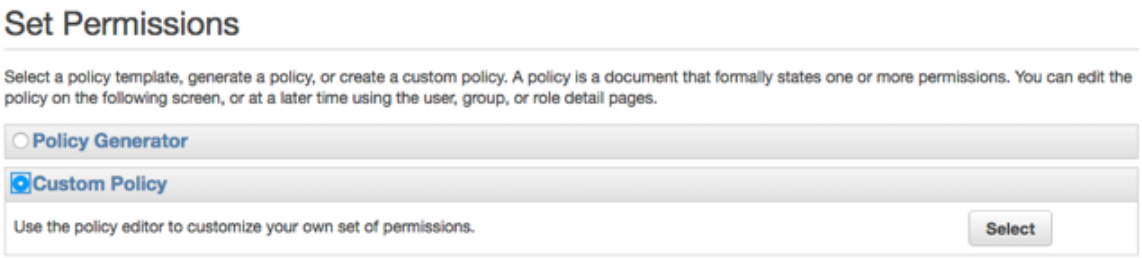

Figure 3.31 - Adding a Custom Policy.

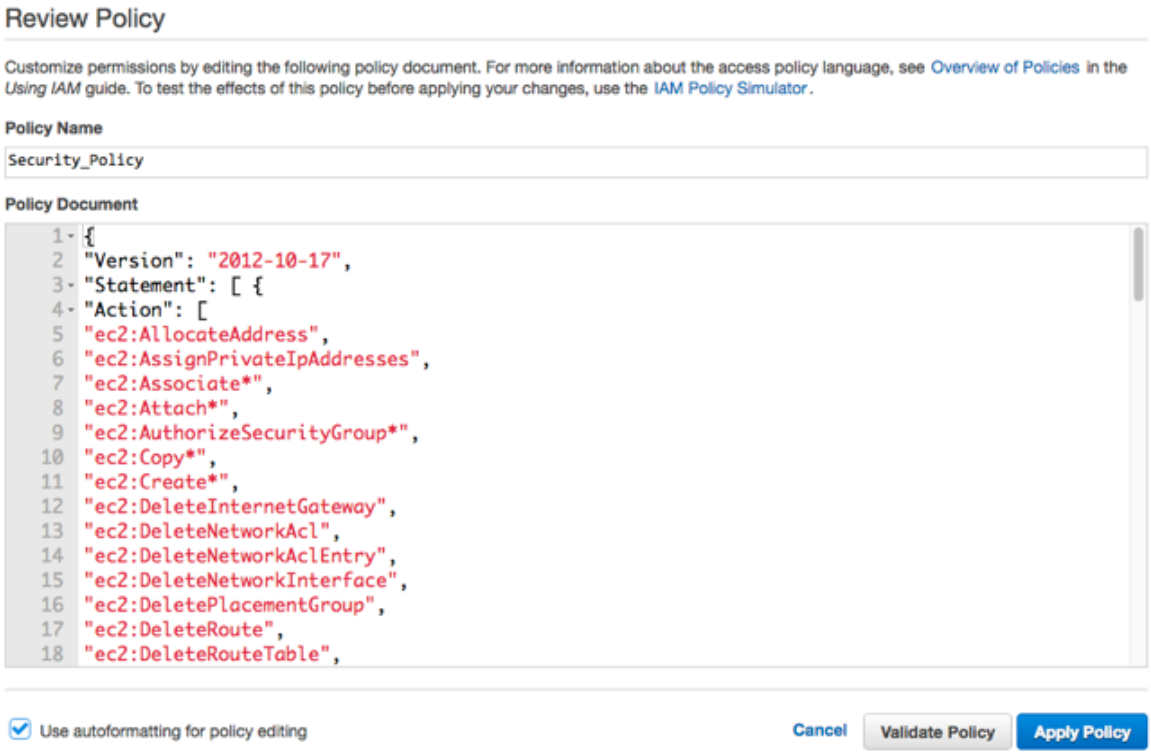

Figure 3.32 - Provide the policy name and contents.

Use the fields on the `Set Permissions` dialog (Figure 3.32) 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 3.33).

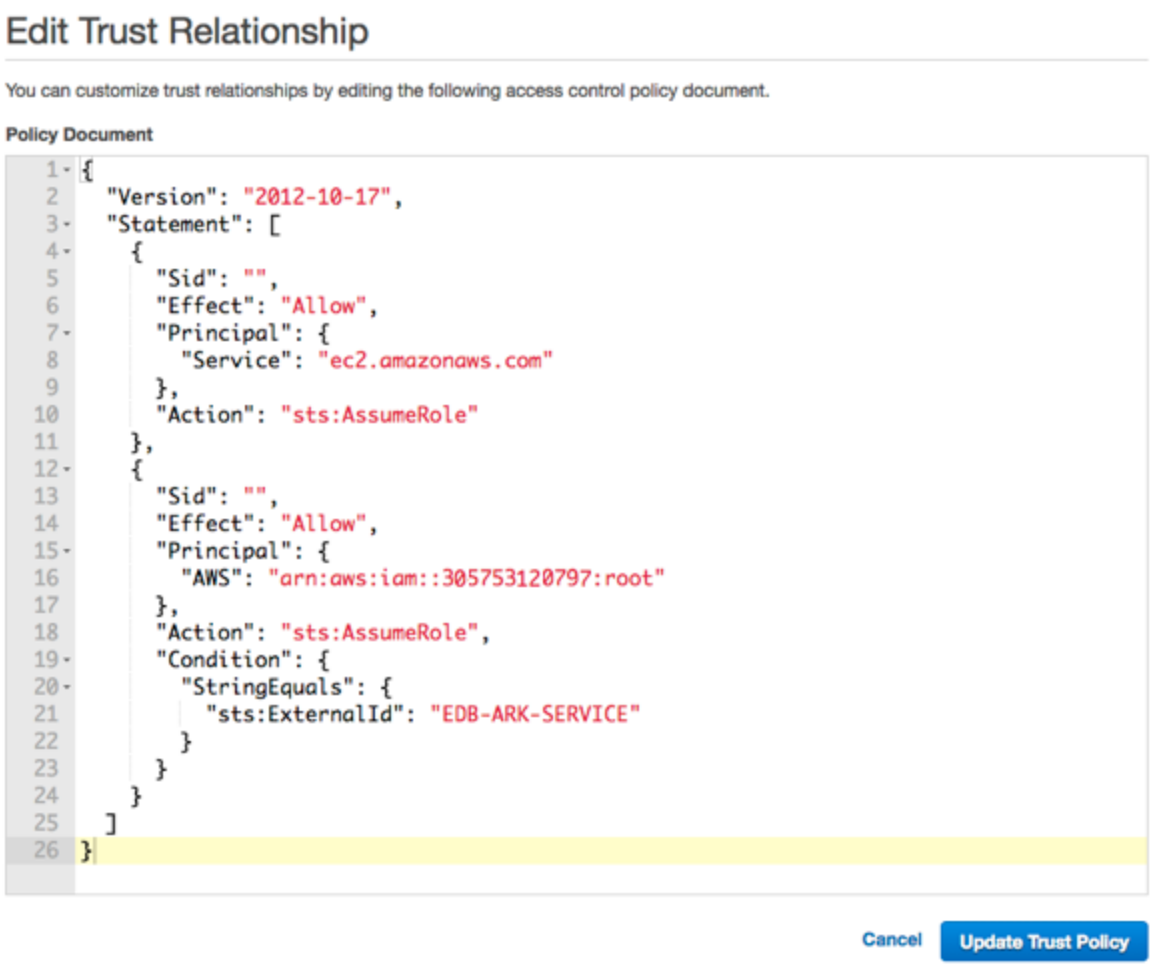

Figure 3.33 - The Policy Document.

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

`EDB-ARK-SERVICE` is a placeholder within the trust policy provided in section 10.4. 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 3.34).

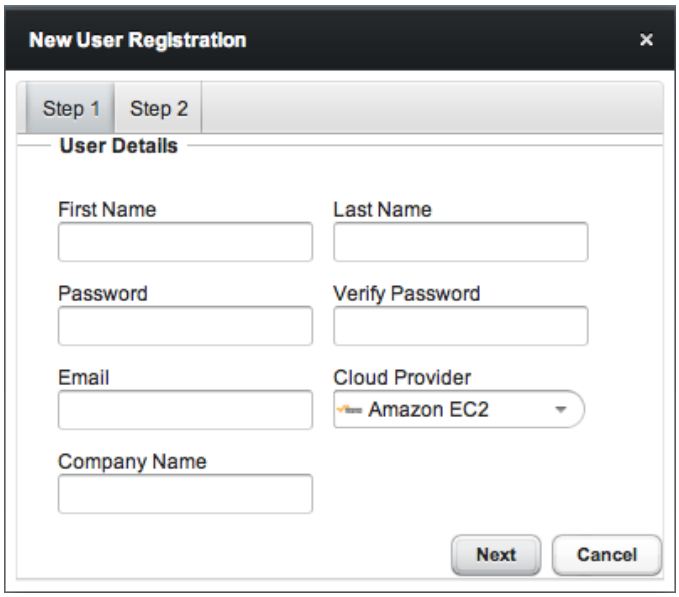

Figure 3.34 - The New User Registration dialog.

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.

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-ARK-SERVICE`. 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.

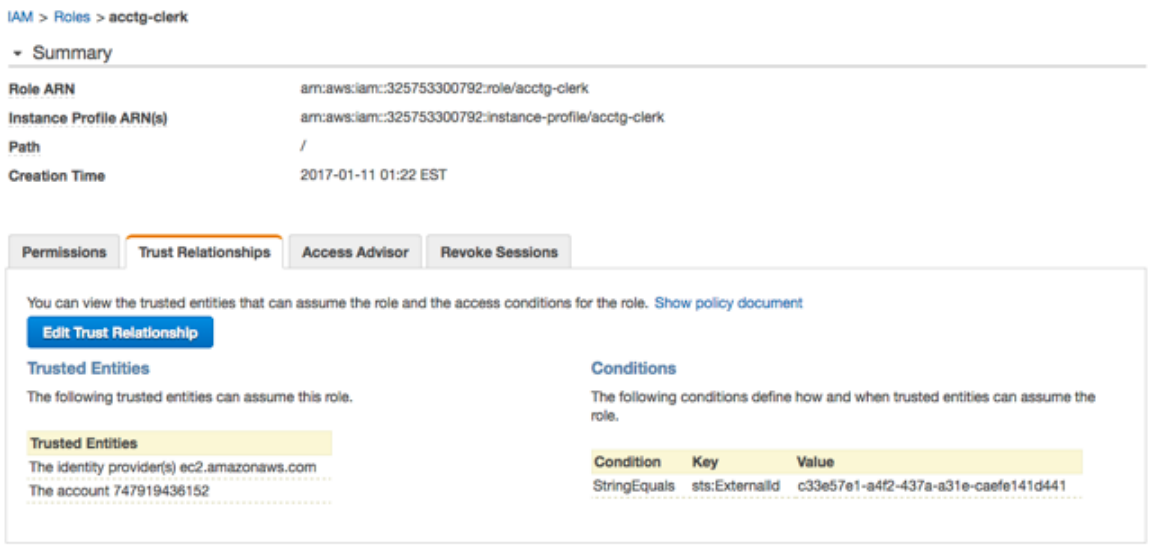

Figure 3.35 - The Summary tab of the Role detail panel.

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 3.35).

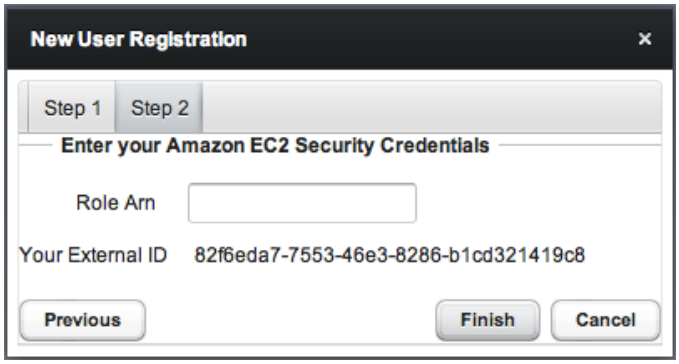

Figure 3.36 - Registering a user on an Amazon EC2 cloud.

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 3.36). 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 3.37).

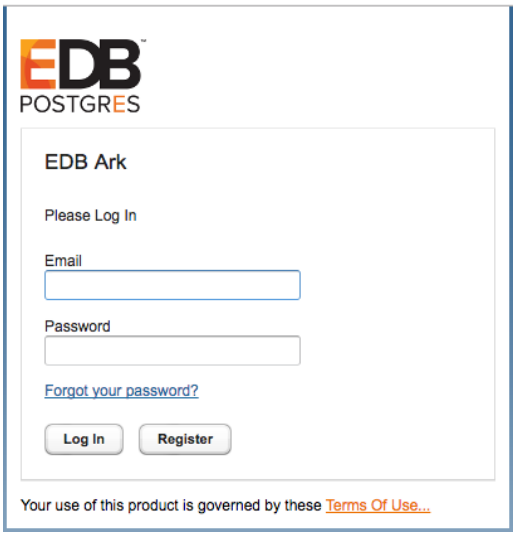

Figure 3.37 - The Login/Register dialog.

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 3.38).

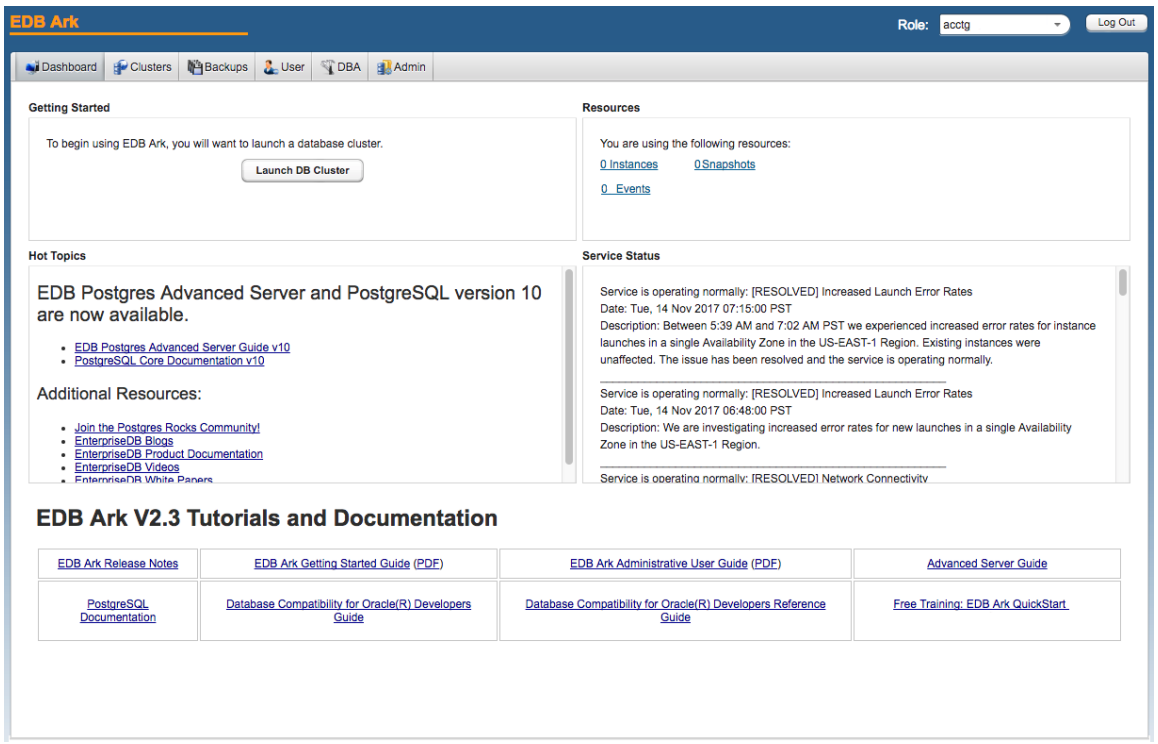

Figure 3.38 - The Dashboard tab of the Ark management console.

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.

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.

2. Create the EDB Ark Security Group.

3. Launch the Ark console instance.

4. Assign a floating IP address to the instance.

5. Complete the Ark console setup dialog, and start the Ark console.

6. Configure OpenStack user accounts.

7. Connect to the Ark console.

You must be an OpenStack administrative user to import and deploy the EDB Ark image, but an OpenStack administrative user may use the *OpenStack Standalone Security Model* to grant access to the Ark Administrator's console to non-administrative OpenStack users. If the OpenStack Standalone Security Model is used, the service account is automatically granted access to the Ark Administrator's console.

The following sections will walk you through the required steps. Please note that during the installation and setup you have the option to create a volume; you should complete the Ark console installation before creating a volume.

3.2.1 OpenStack Prerequisites

The following sections note the prerequisite steps required to install and run EDB Ark on an OpenStack host.

Enabling the Keystone Identity Service 2.0 API

By default, OpenStack Mitaka, Newton, and Ocata enable the Keystone identity service version 3.0 API; version 3.0 is not supported by EDB Ark. Before using EDB Ark on an OpenStack host that uses version 3.0, 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`.

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
```

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.

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.

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 2.12).

Figure 2.12 – A resource limit error.

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

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

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

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

delete the backup. Deleting backups of obsolete clusters will free up system resources for use.

Configuring Ark on an HTTPS Enabled Host

If your Ark console resides on an OpenStack host that enables HTTPS endpoints, you must import the OpenStack SSL certificates to the Ark's Glassfish web server. Please note that you must import the certificates immediately after the Ark instance is started, and before configuring the console. For detailed information about importing the SSL certificates, please see Section 6.5.

3.2.2 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.

Please note: the photos and descriptions in the following sections use screenshots and descriptions from the OpenStack Ocata administrator's console.

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 3.39).

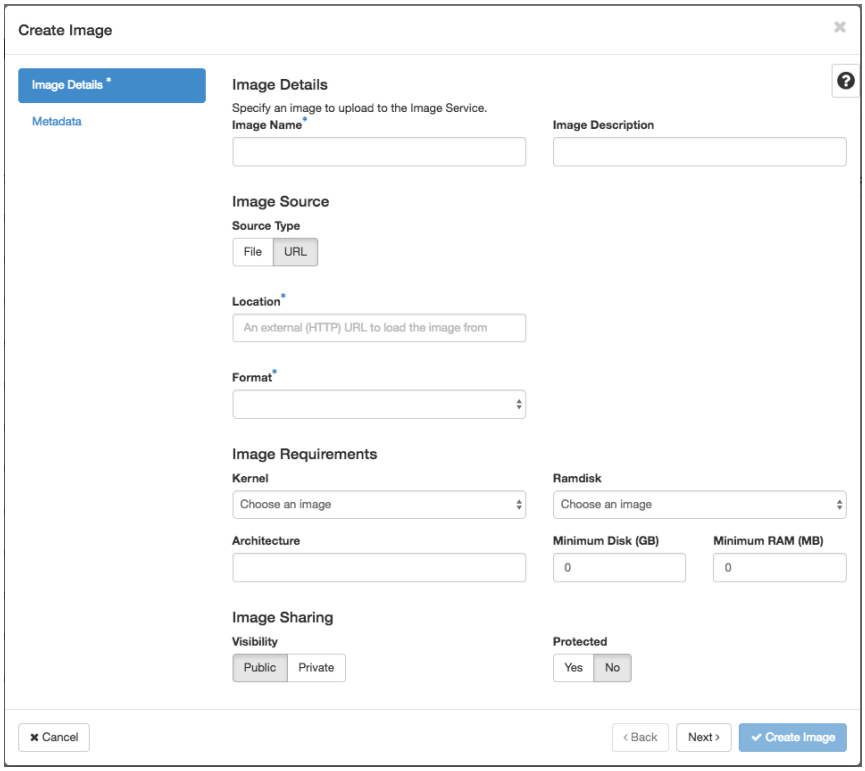

Figure 3.39 – The Create Image dialog.

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

- Use the `Image Name` field to provide a name for the image.
- Use the `Image Description` field to provide a description of the image.
- Use the `Image Source` selector to specify that the source will be an `Image File`.
- Use the `Location` field to specify the location from which the image will be loaded.
- 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 `Minimum RAM (MB)` field.
- Use the `Visibility` selector to specify if the image is `Public` or `Private`.
- Set the `Protected` field to `Yes` to indicate that the image may only be deleted by a user with permissions can delete the image.

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
    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/

3.2.3 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. Expand the `Project` menu, and the `Network` menu; then, select `Security Groups`.

4. Click the `+ Create Security Group` button to open the `Create Security Group` dialog (shown in Figure 3.40).

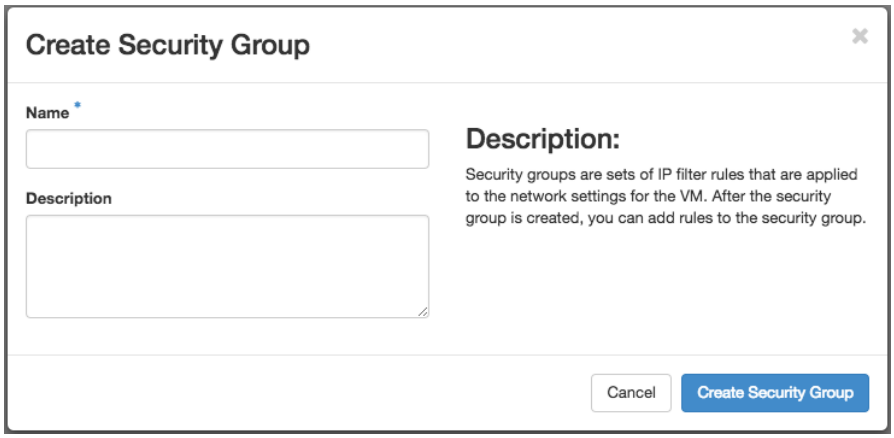

Figure 3.40 – The Create Security Group dialog.

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

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

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

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

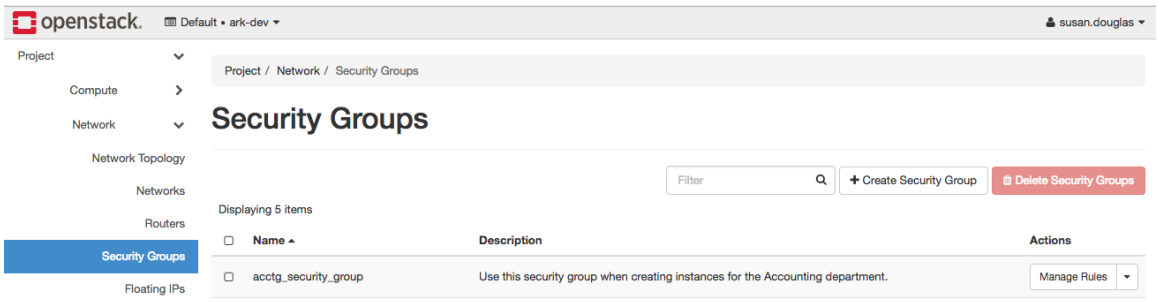

Figure 3.41 – The new group, displayed in the Security Groups list.

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 3.41). When the list of security group rules opens (see Figure 3.42), click the `+ Add Rule` button to access a dialog that allows you to add a new rule.

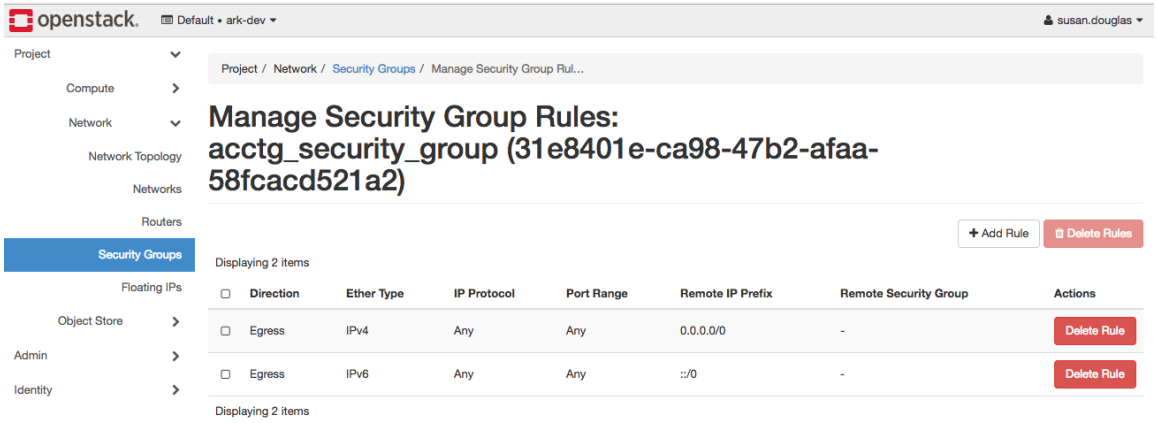

Figure 3.42 – *The new security group.*

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

3.2.4 Launching the EDB Ark Console Instance

After importing the image and defining the security group, you are ready to launch the Ark console instance. The instructions that follow list the selections that are required to launch an Ark console instance on an OpenStack Ocata host. The configuration of your host may require you to provide additional system-specific information; please consult the OpenStack documentation for your version.

To access a list of instances, open the `Project` menu, then the `Compute` menu and select `Instances`. To create a new instance, click the `Launch Instance` button to open the `Launch Instance` dialog (shown in Figure 3.43).

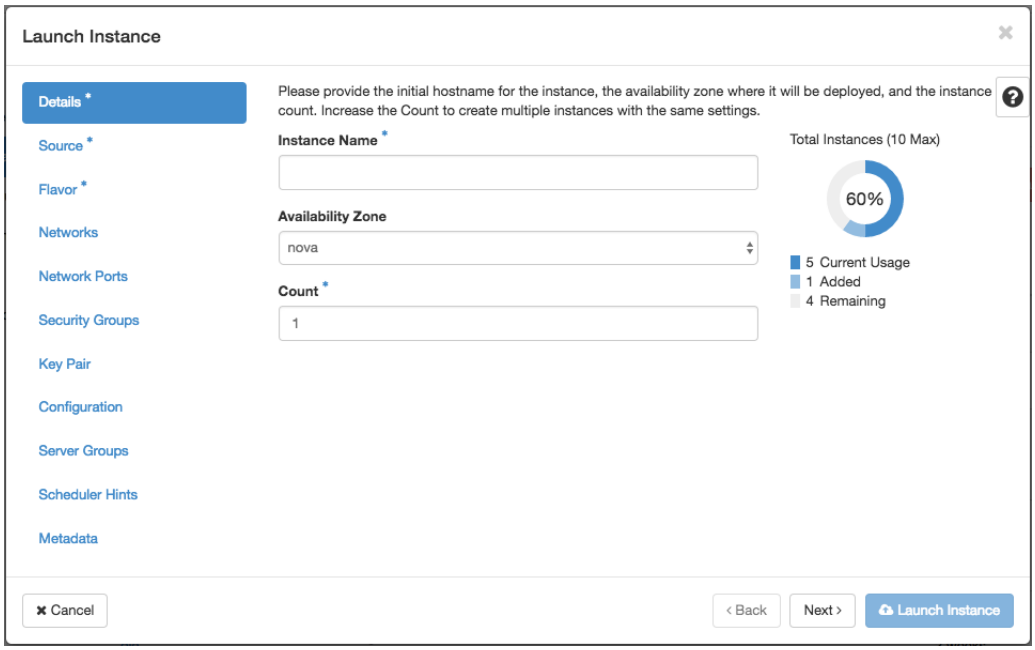

Figure 3.43 – *The Launch Instance dialog.*

Use fields on the `Launch Instance` dialog to describe the EDB Ark instance; on the `Details` tab:

- Use the `Instance Name` field to provide a name for the instance.
- Use the `Availability Zone` drop-down listbox to specify an availability zone.
- Set the `Count` field to `1`.

On the `Source` tab:

- Use the drop-down listbox in the `Select Boot Source` field to select `Image`.
- Set the `Create New Volume` selector to `No`.
- Click the up arrow to the right of an instance name to add the image name to the `Allocated` field; this selects the image as the backing image for the instance.

On the `Flavor` tab:

- Click the up arrow to the right of an image description to move the description to the `Allocated` field; this selects the instance size.

On the `Networks` tab:

- Select a network from the list of available networks.

No changes are required on the `Network Ports` tab.

On the `Security Groups` tab:

- Click the up arrow to the right of a security group name to move the description to the `Allocated` field; this selects the security group that will be applied to the instance.

On the `Key Pair` tab:

- Click the `Create Key Pair` button to open a dialog that allows you to create a new keypair or click the `Import Key Pair` button to open a dialog that allows you to select an existing key pair.
- Click the up arrow to the right of a `Key Pair` name to select the keypair you will use to access the instance.

On the `Configuration` tab:

- Use the `Customization Script` field to provide a script that sets a password for the console setup dialog:

```
#!/bin/sh
rm -f /var/ppcd/startup-password.txt
echo "console_password" > /var/ppcd/startup-password.txt
chown ppcd:ppcd /var/ppcd/startup-password.txt
chmod 600 /var/ppcd/startup-password.txt
```

Where *console_password* specifies the password that allows access to the setup dialog.

When the first user connects to the AWS Ark console, they will be required to provide the *console_password* provided in the script; entering the password will invoke the setup dialog.

console_password is stored in /var/ppcd/startup-password.txt.

No changes are required on the Server Groups tab:

No changes are required on the Scheduler Hints tab:

No changes are required on the Metadata tab:

Click the Launch Instance button to launch the console instance. When OpenStack finishes creating the instance, it will be displayed in the list on the Instances window.

3.2.5 Assign a Floating IP Address

When the instance launch completes, the new instance will be displayed on the `Instances` panel (as shown in Figure 3.44).

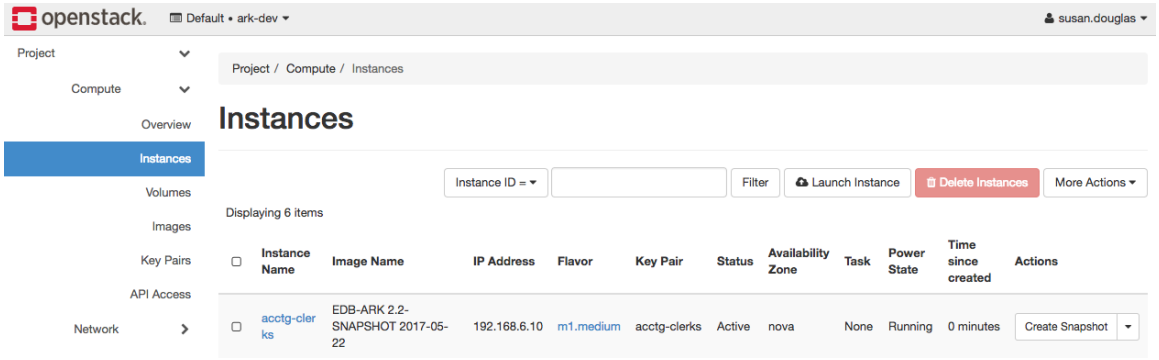

Figure 3.44*– The Instances dialog.*

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

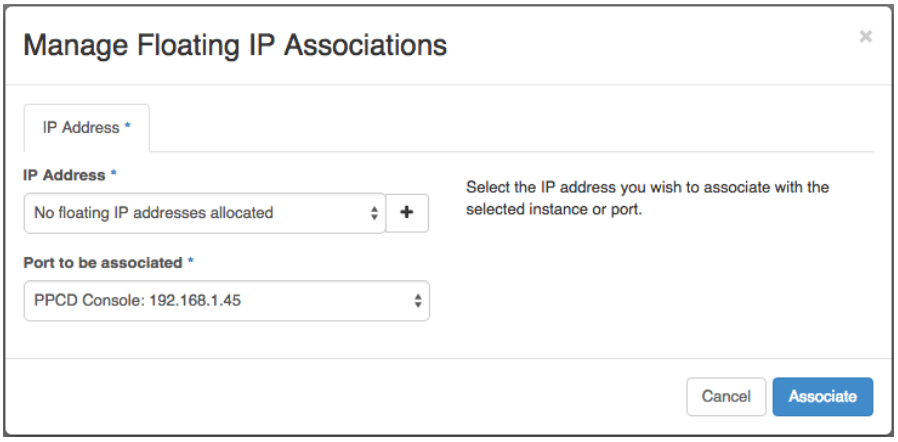

Figure 3.45*– The Manage Floating IP Associations dialog.*

3.2.6 Deploying the Ark Console

To access the Ark setup dialog and configure the console, open a browser window and navigate to the floating IP address assigned to the console.

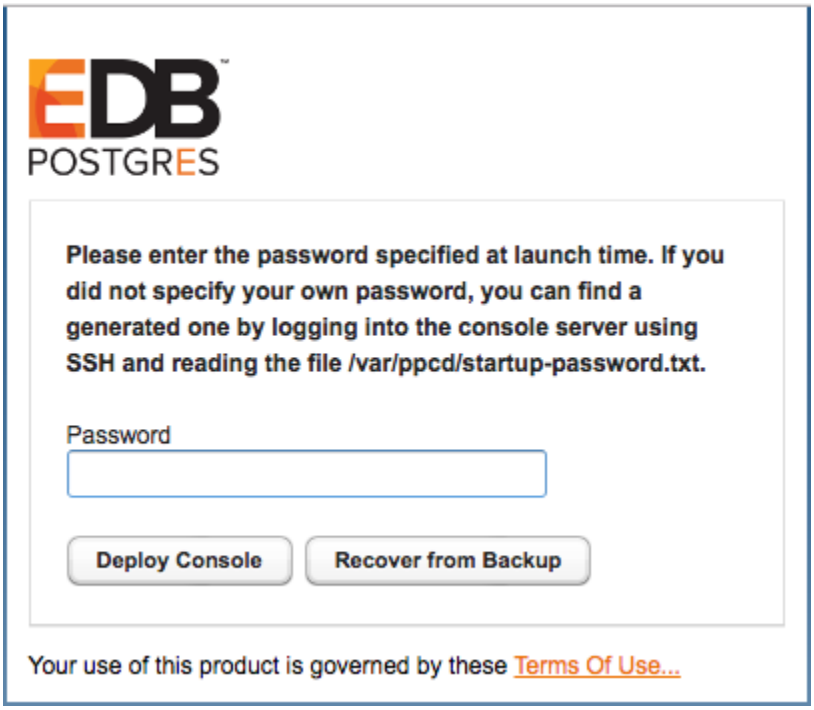

Figure 3.46 – Logging in to the instance.

When prompted, provide the password specified when launching the console in the `Password` field and click `Deploy Console`. The Ark console setup dialog opens as shown in Figure 3.47.

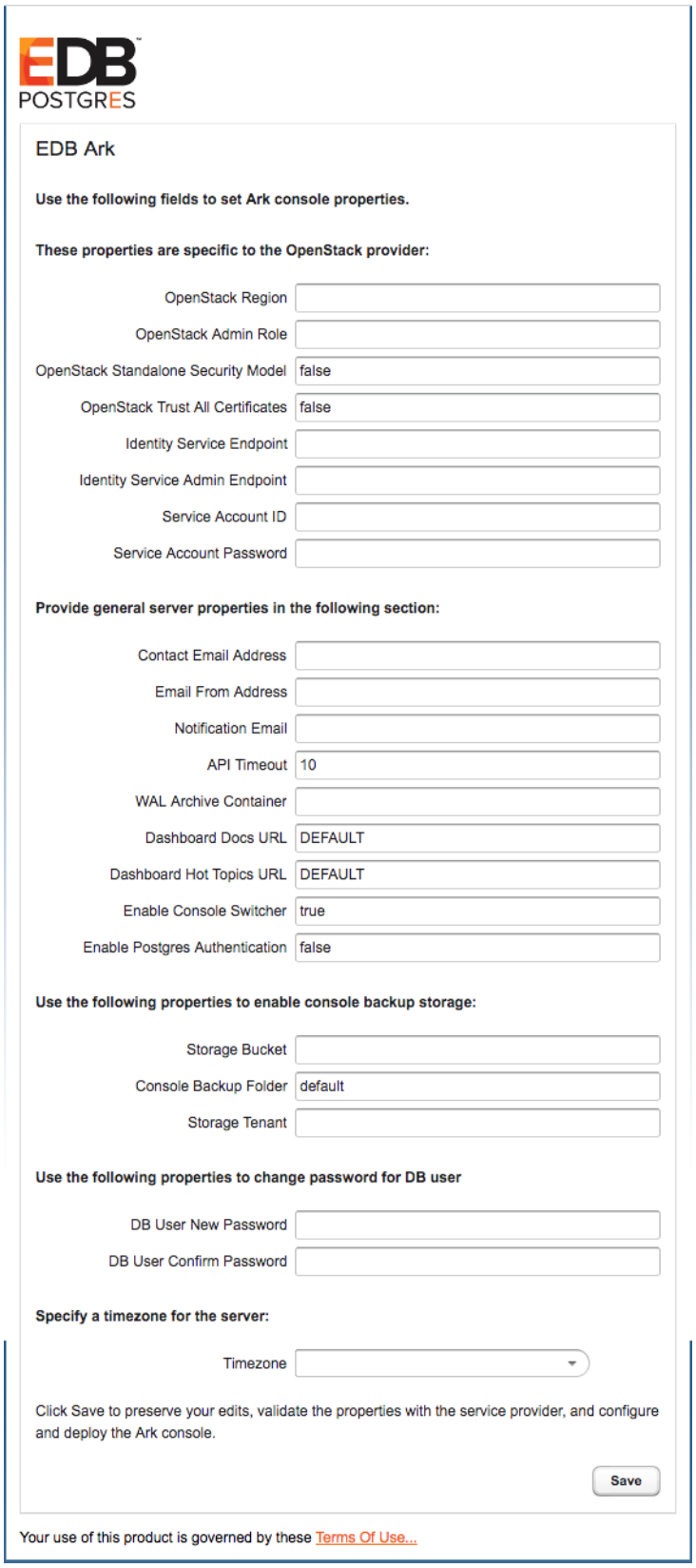

Figure 3.47 – Configuring the Ark console.

Use fields on the setup dialog to provide provider specific information and configuration details for the Ark console. The fields in the first section of the setup dialog set values that are OpenStack specific; provide values that match the values specified in your OpenStack management console:

- Use the `OpenStack Region` field to specify the region in which the OpenStack host resides.

- Use the `OpenStack Admin Role` field to specify the name of the OpenStack administrative role. When a user that is a member of this role connects to the console, the console will display the Ark administrative console (which includes the `Admin` and `DBA` tabs).

- Use the `OpenStack Standalone Security Model` field to instruct Ark to allow an OpenStack administrator to grant access to the Ark administrative console to non-administrative OpenStack users.

  If you specify `true`, Ark will evaluate the `clouduser` table in the backing `postgres` database to determine if the user should have Ark administrative access. When the `OpenStack Standalone Security Model` field is `true`, the Administrative console will display the `User Administration` table, which allows an OpenStack administrator to manage user privileges.

  If the `OpenStack Standalone Security Model` field is `false`, the `Service Account ID` must be an OpenStack administrative user; the Ark administrative console access will not be enabled for non-administrative users.

- Specify `true` in the `OpenStack Trust All Certificates` field to disable SSL checks by the Ark console.

- Use the `Identity Service Endpoint` field to specify the URL of the OpenStack Keystone Identity Service.

- Use the `Identity Service Admin Endpoint` field to specify the URL of the OpenStack Keystone Administrative Service.

- Use the `Service Account ID` field to specify the name of the OpenStack user account that Ark will use when managing clusters. The account must be a member of and be assigned the `admin` role for all tenants that are allowed to run Ark clusters.

- Use the `Service Account Password` field to specify the password associated with the OpenStack service account.

The fields in the general properties section to set values that control Ark behaviors:

- Use the `Contact Email Address` field to specify the address that will be included in the body of cluster status notification emails.

- Use the `Email From Address` field to specify the return email address specified on cluster status notification emails.

- Use the `Notification Email` field to specify the email address to which email notifications about the status of the Ark console will be sent.

- Use the `API Timeout` field to specify the number of minutes that an authorization token will be valid for use within the API.

- Use the `WAL Archive Container` field to specify the name of the storage container where WAL archives (used for point-in-time recovery) are stored. You must provide a value for this property; once set, this property must not be modified.

- Use the `Dashboard Docs URL` field to specify the location 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 parameter to `DEFAULT` to display content (documentation) from EnterpriseDB; to display alternate content, provide the URL of the content. To display no content in the lower half of the `Dashboard` tab, leave the field blank.

- Use the `Dashboard Hot Topics URL` field to specify the location 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 parameter to `DEFAULT` to display content (alerts) from EnterpriseDB; to display alternate content, provide the URL of the content. To display no content across the middle section of the `Dashboard` tab, leave the field blank.

- Use the `Enable Console Switcher` field to indicate if the console should display console switcher functionality. When set to `true`, the console will display the switcher; when `false`, the switcher will not be displayed. For more information, see Section 4.1.1.

- Set `Enable Postgres Authentication` to `true` to instruct Ark to enforce the authentication method configured on the backing Postgres server. Supported authentication methods include password, LDAP, RADIUS, PAM, and BSD.

  If `false`, Ark will use the default authentication method (password).

If you are using a Swift Object Storage service with your OpenStack installation, provide information about the location of the console backup storage in the next section of the

setup dialog; please note that you must provide values in these fields to use the Ark console recovery functionality:

- Use the `Storage Bucket` field to specify the name of the container that will be used to store files for point-in-time recovery. This location should not change after the initial deployment of the Ark console.

- Use the `Console Backup Folder` field to specify a folder in which the backups will be stored.

- Use the `Storage Tenant` field to provide the name of the tenant in which the backup will be stored.

Use the password properties fields to modify the password for the database user:

- Use the `DB User New Password` field to modify the database password.

- Use the `DB User Confirm Password` field to confirm the new password.

Use the last field to specify a timezone for the server:

- Use the drop-down listbox in the `Timezone` field to select the timezone that will be displayed by the Ark console.

When you've completed the console properties dialog, click the `Save` button.

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 allow access to an OpenStack project, connect to the OpenStack console as an Administrative user and expand the `Identity` menu; then click the `Projects`. To modify the access privileges for a project, use the drop-down listbox in the `Actions` column to the right of the project name to select `Modify Groups`.

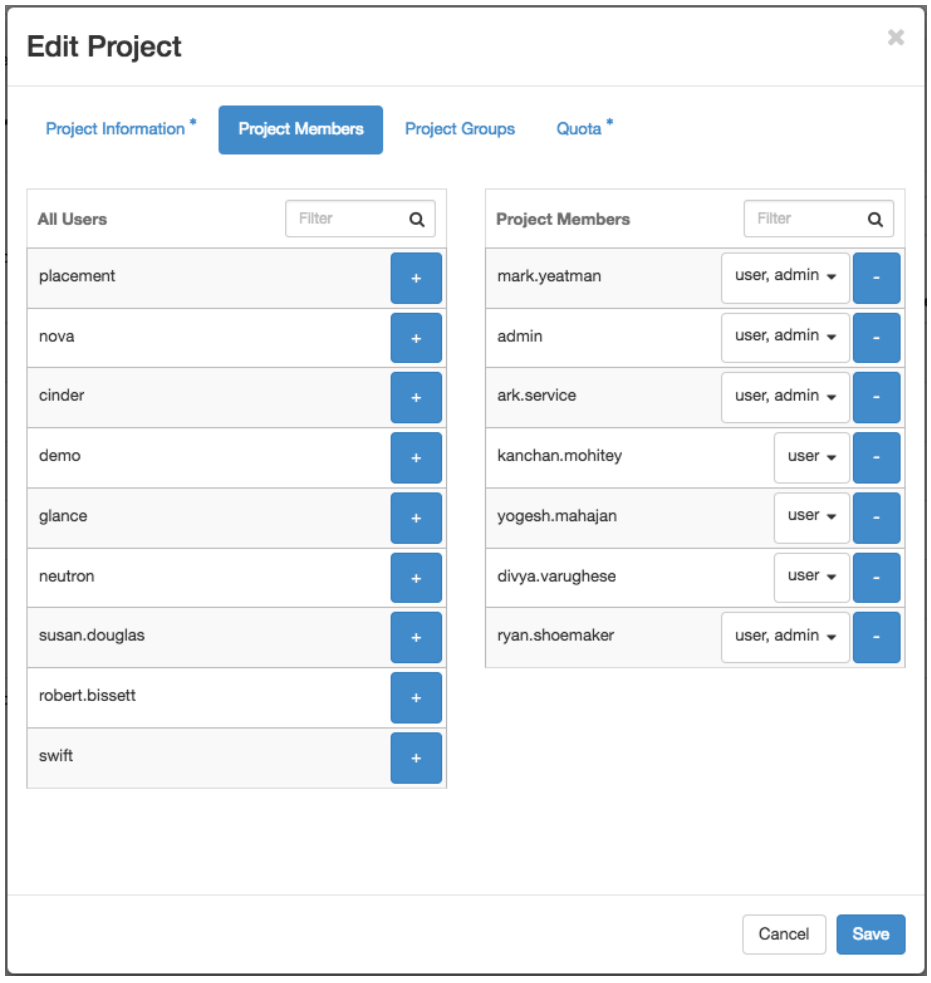

Figure 3.48 – The Edit Project dialog.

When the `Edit Project` dialog opens, navigate to the `Project Members` tab (see Figure 3.48).

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

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

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

Creating an OpenStack User with EDB Ark Console Access

To create an OpenStack user account with access to the Ark console for a specific project, connect to the OpenStack console as a user with Administrative privileges and select `Identity`. Open the `Users` tab, and click the `Create User` button to open the `Create User` dialog (see Figure 3.49).

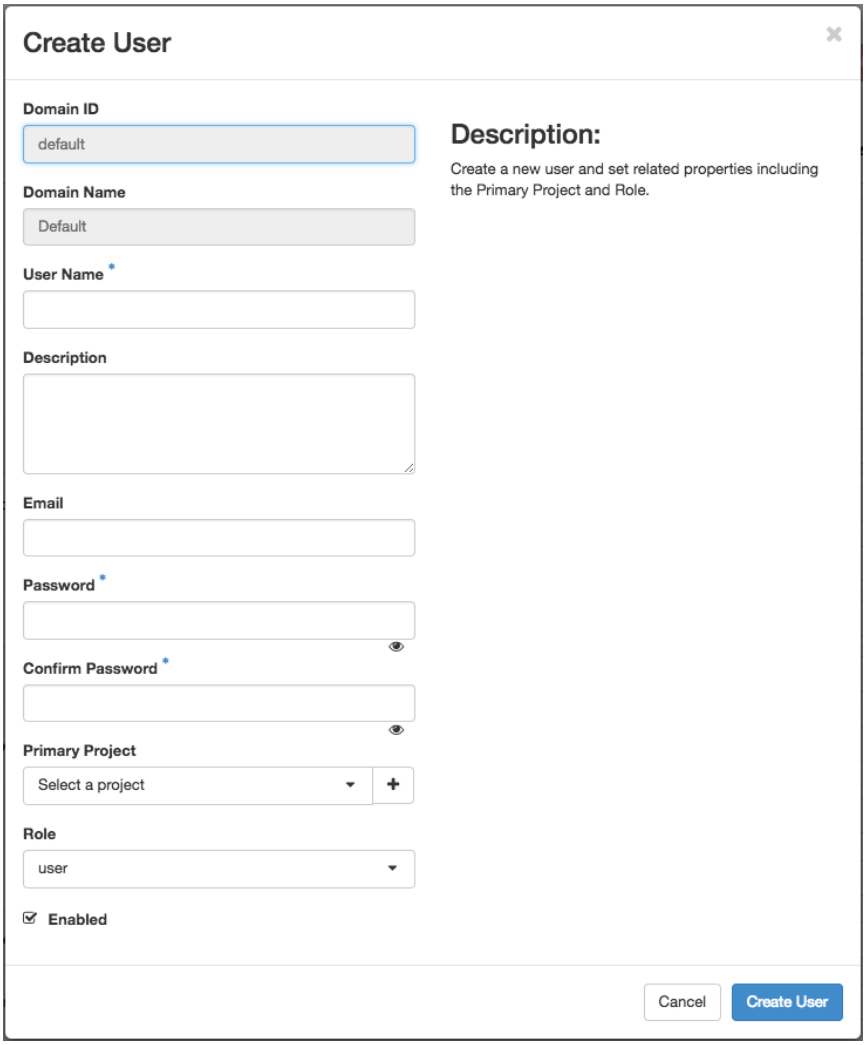

Figure 3.49 - The Create User dialog.

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

- If applicable, provide the identifier for the domain in the Domain ID field.
- If applicable, provide the name of the domain in the Domain Name field.
- Specify the name of the user in the User Name field.
- Provide descriptive text or comments in the Description field.
- Specify the email address of the user in the Email field.
- Specify the password associated with the user account in the Password field.
- Re-enter the password in the Confirm Password field.
- Use the drop-down listbox in the Primary Project field to select the project that will be displayed when the user connects. Please note that the Ark service account must have administrative privileges in the selected project.
- Use the drop-down listbox in the Role field to specify if the new role is a user or an admin user. Please note that user roles will have sufficient privileges to access the Ark console.
- If the new user is currently active and should be allowed to access the project selected, check the box next to Enabled.

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

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 3.50).

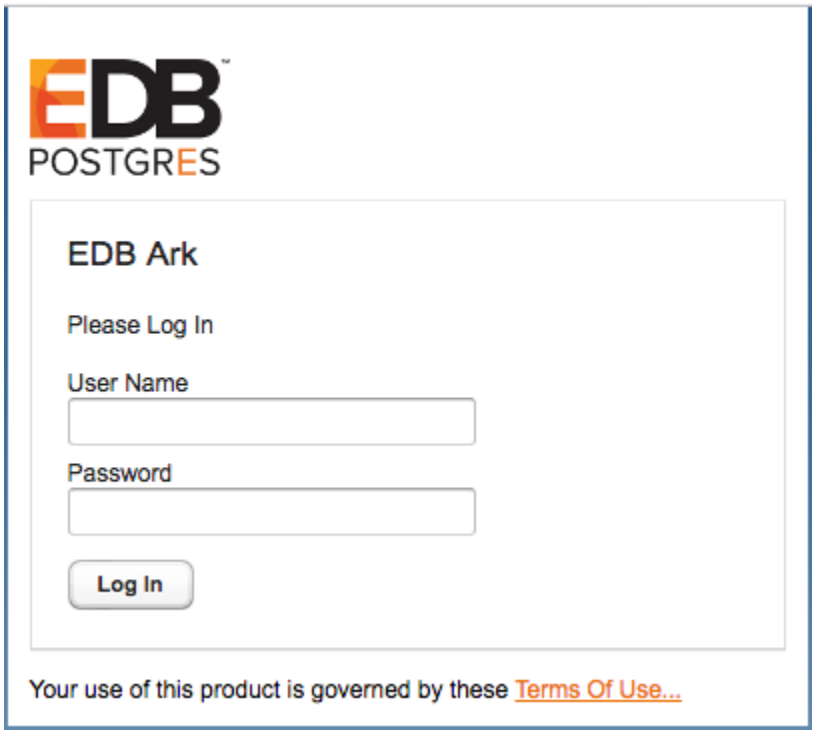

Figure 3.50 - The Login dialog.

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 3.51).

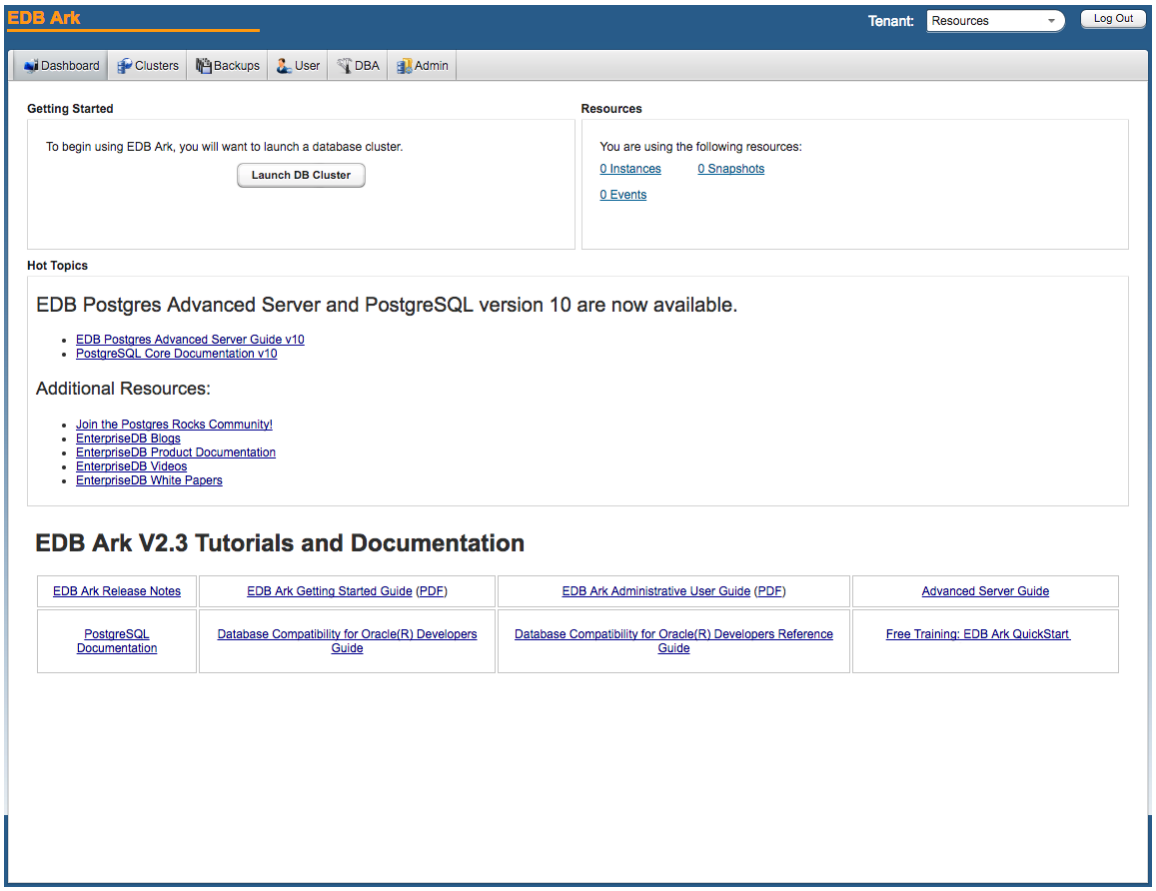

Figure 3.51 - The EDB Ark Administrator's console.

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.

3.3 Installing EDB Ark for Azure

The EDB Postgres Ark image is available on Azure Marketplace; installation and configuration is a simple process. To enable the Ark console on Azure, you must:

- Create an Azure user account with sufficient privileges to access the Ark Administrator's console. For more information, see Section 3.3.1.
- Create an Azure network security group. For more information, see Section 3.3.2.
- Create an Azure storage account. For more information, see Section 3.3.3.
- Launch a VM Image that contains the Ark console. For more information, see Section 3.3.4.
- Configure the Ark console. For more information, see Section 3.3.5.
- Register an Ark Administrative user. For more information, see Section 3.3.6.

3.3.1 Providing Administrative Access to an Azure User

To provide sufficient privileges for an Azure user account to access the Ark administrative console, navigate to the Azure `Resource groups` panel, highlight the name of the resource group in which your instance will reside, and select `Access control (IAM)` from the `Resources` panel; then, click the `+Add` button to access the `Add permissions` panel.

On the `Add permissions` panel, use the drop-down listbox in the `Role` field to select `Owner`; use the drop-down listbox in the `Select` field to select the user(s) that should have administrative access to the Ark console. When you've made your selections, click `Save`.

To limit the `Scope` of the access to the resource group in which the image resides, use the `Resources – Access control (IAM)` panel to specify a value of `This resource` in the `Scope` field for the specified user(s).

For more information about delegating Azure permissions, please visit:

https://docs.microsoft.com/en-us/azure/active-directory/role-based-access-control-configure

3.3.2 Creating a Security Group

Before connecting to the Ark console, you must create a security group that will allow connections from your web browser, and between the Ark console and your instance. To create a security group, navigate to the Microsoft Azure Network security groups page, and click the Add button. When the Create network security group panel opens:

- Use the Name field to provide a name for the security group.
- Use the drop-down listbox in the Subscription field to select a subscription plan.
- Use the Resource group field to provide a name for the associated resource group, or highlight the Use existing radio button and use the drop-down listbox in the Resource group field to select an existing resource group.
- Use the Location drop-down listbox to specify a location.

When you've finished, click Create to create a network security group.

After creating the network security group, you must provide the inbound rules that will allow the Ark console to manage your instance. On the Network security groups page, click the name of the security group that you wish to modify; click Inbound security rules (in the SETTINGS section of the details panel) to modify the inbound rules for the group.

To add a new rule, click the Add button, and provide details about the rule; after providing rule details, click OK. The Azure console will notify you that it is creating the new rule. When defining the security group, include the rules listed below:

Rule Type	Direction	Port	Remote	CIDR Address
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

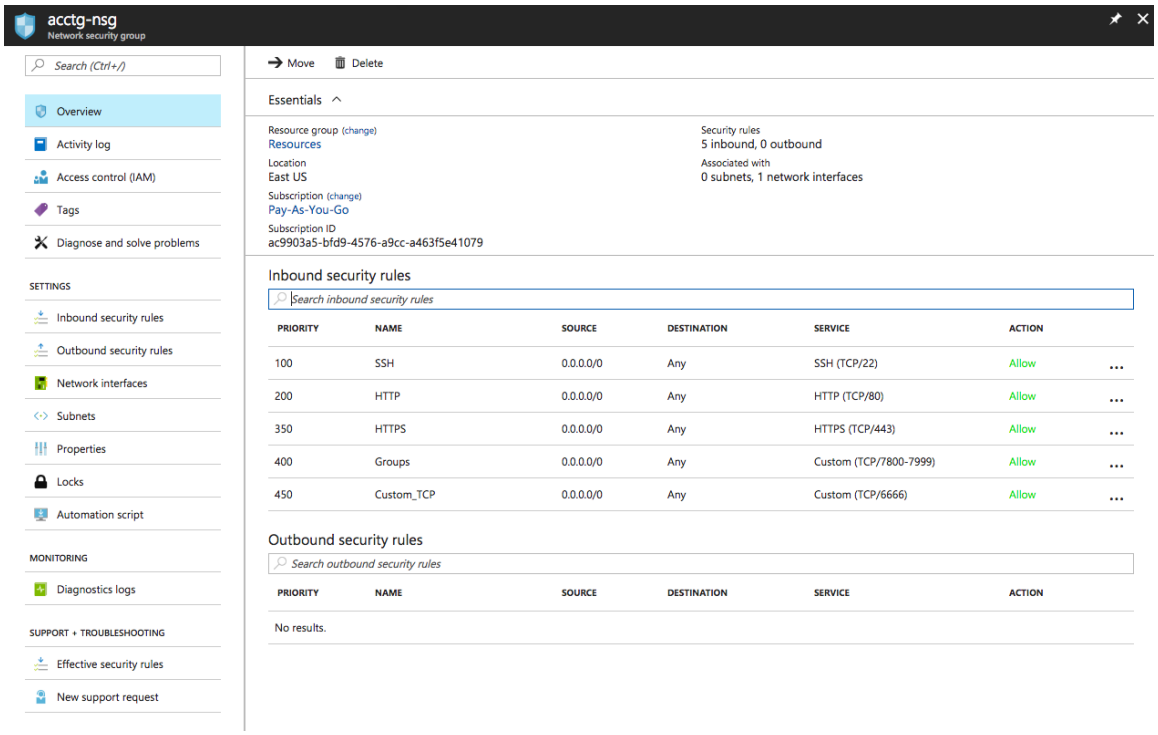

Figure 3.52 – Reviewing security group rules.

Select `Overview` to review the rules defined for a security group (see Figure 3.52).

The CIDR addresses specified in the rules for SSH, HTTP, and HTTPS 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 rule that opens ports `7800` through `7999` provides enough ports for 200 cluster connections; you can extend the upper limit of the port range if more than 200 clusters are required.

3.3.3 Creating a Storage Account

Before launching the Ark console, you should create an Azure storage account in which the Ark console will store console backups. You should not modify the storage account after the console is launched.

To add an Azure storage account, navigate to the Azure `All resources` page, and click the `Add` button. In the `MARKETPLACE` edit box enter `Storage account`, and hit return. Highlight the `Storage account – blob, file, table, queue` entry.

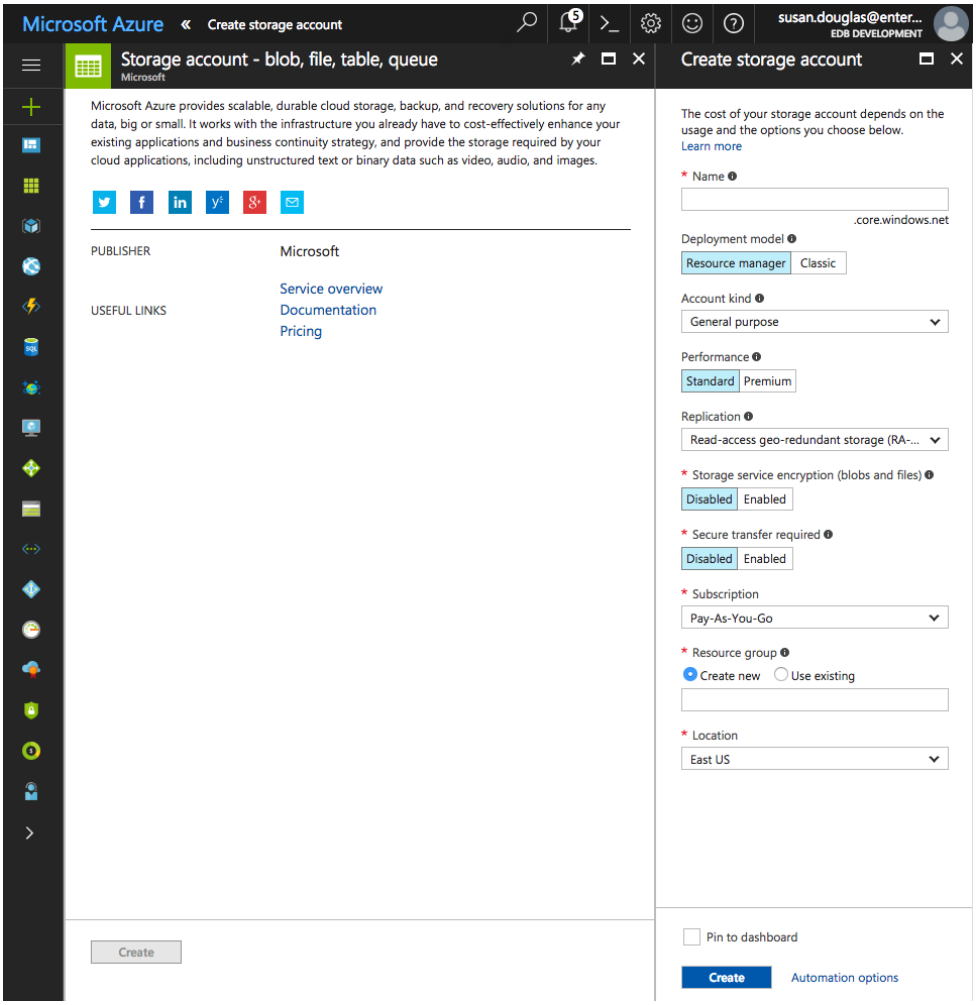

Figure 3.53 – Defining a storage account.

Click the `Create` button located on the bottom of the `Storage account-blob, file, table, queue` panel to open the `Create storage account` panel. Use fields on the `Create storage account` panel to define the storage account (see Figure 3.53).

When you've defined your storage account, click `Create`; the Azure dashboard will keep you informed as the storage account is deployed, and send you a notification when the account creation is finished.

For detailed information about defining a storage account, please see the Azure documentation at:

https://docs.microsoft.com/en-us/azure/storage/

3.3.4 Launching the Ark Console Instance

The EDB Postgres Ark image is available on the Microsoft Azure Marketplace. To create an Ark virtual machine, log in to the Microsoft Azure management console, and click the green plus sign in the upper-left hand corner to navigate to the Azure Marketplace.

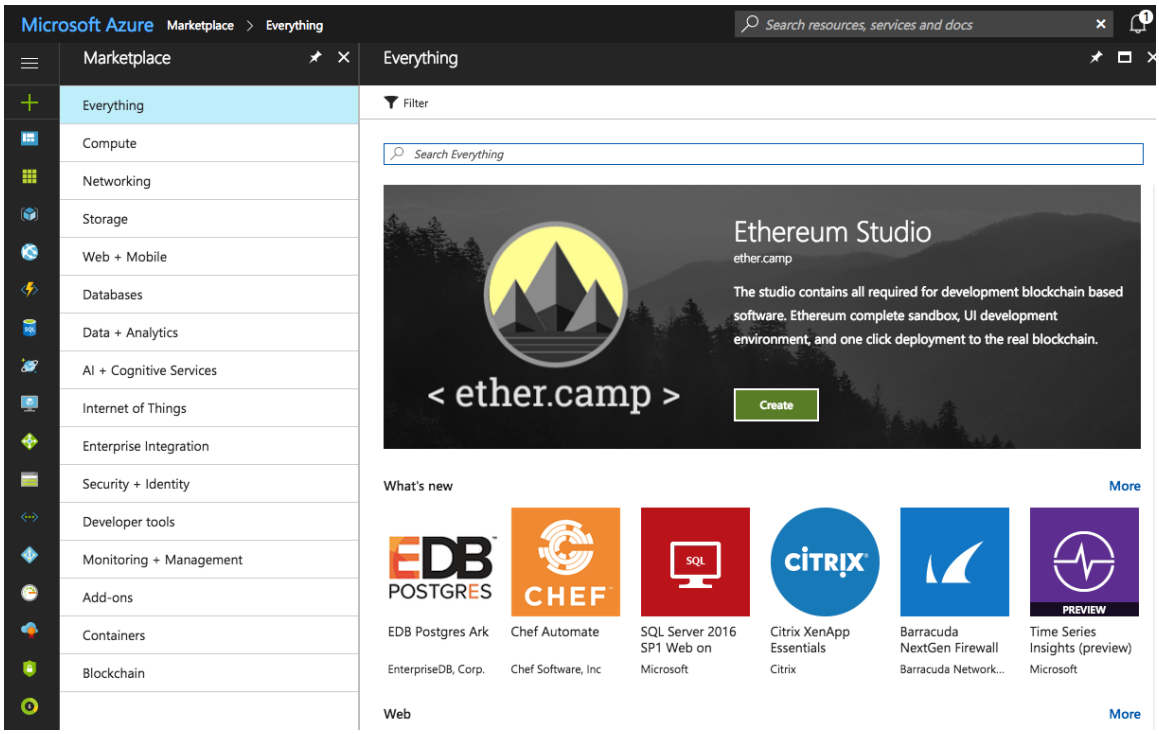

Figure 3.54 – Selecting an image.

When the Azure Marketplace opens, enter `EDB Postgres Ark` in the search box. Select the `EDB Postgres Ark` (published by EnterpriseDB Corp.) icon from the search results, and click `Create` to continue.

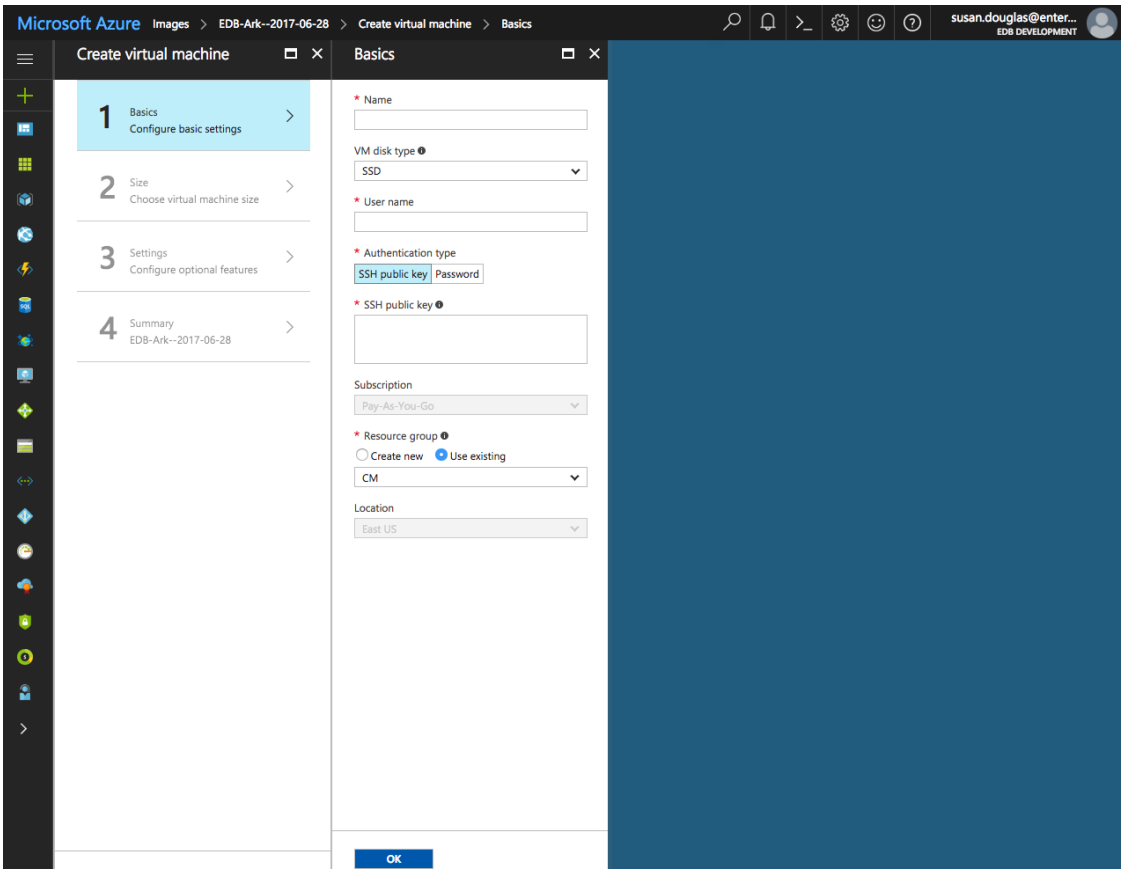

Figure 3.55 – The Basics panel.

Use fields on the `Basics` panel (see Figure 3.55) to provide general information about the new VM:

- Provide a name for the VM in the `Name` field.
- Use the `VM disk type` field to select the disk type for the root volume.
- Provide an operating system user name in the `User name` field.
- Use the `Authentication type` switch to select an authentication type.
- If you elect to enable SSH public key authentication, provide the key in the `SSH public key` field.
- If applicable, use the `Resource group` field to specify a resource group.
- If prompted, use the `Location` field to specify a location.

Click `OK` to continue.

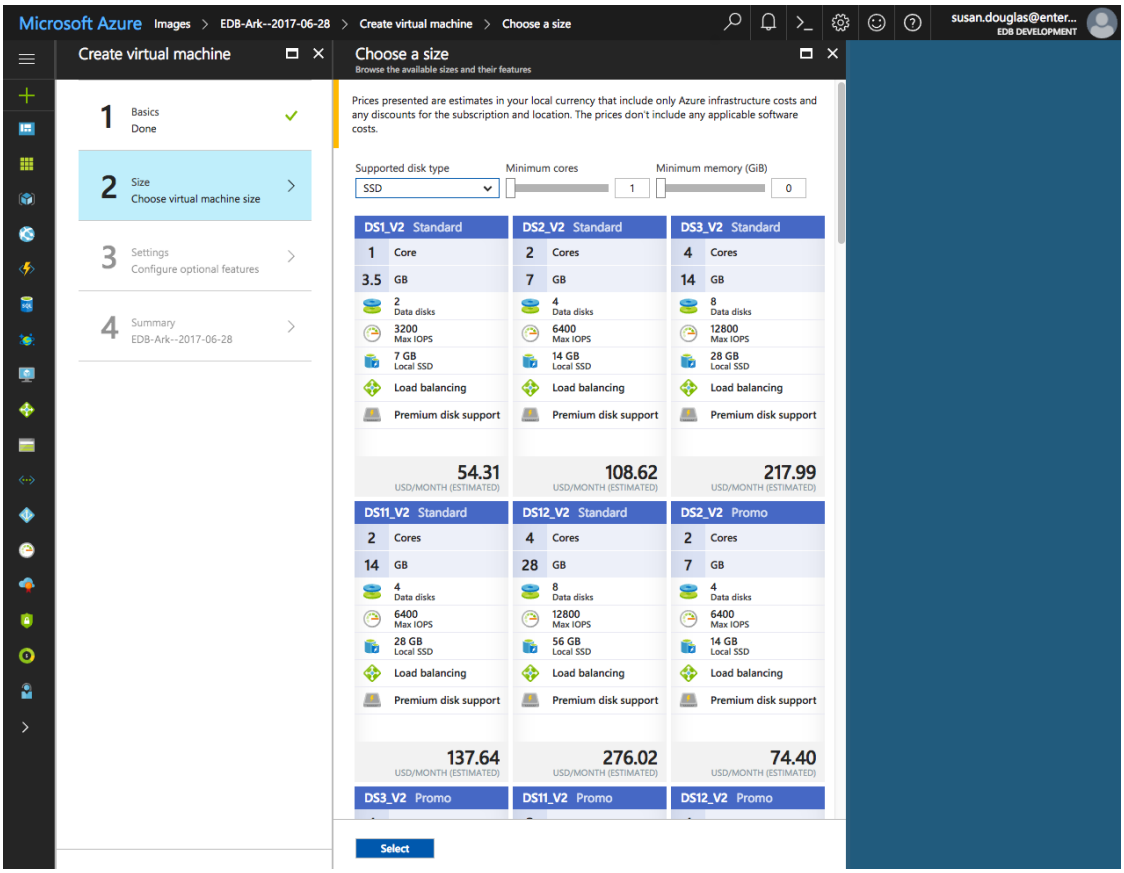

Figure 3.56 – The Size panel.

Use options on the `Size` panel (see Figure 3.56) to specify your preferences about the size of the VM instance:

- Use the `Supported disk` type drop-down listbox to select the disk type for the machine.
- Use the `Minimum cores` slider to specify the minimum number of cores allotted for the machine.
- Use the `Maximum memory` slider to specify the maximum memory allotted for the machine.
- Select a disk size from the disk descriptions shown in the bottom of the page; highlight a disk description to select that size for deployment.

Click `Select` to continue.

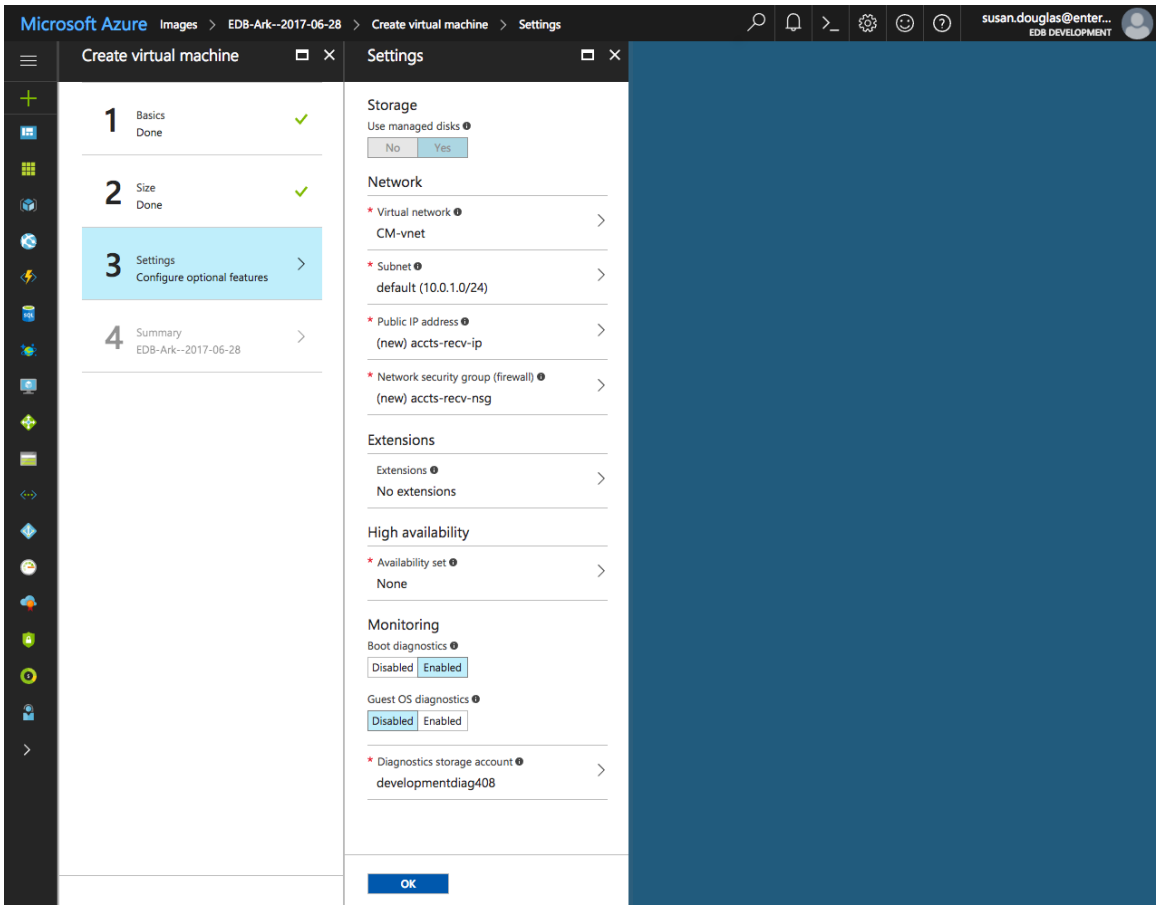

Figure 3.57 – The Settings panel.

Use fields on the `Settings` panel (see Figure 3.57) to specify your configuration preferences for the virtual machine. When configuring an Azure virtual machine to use the Ark console, you should:

- Open the `Network security group` pane and select the security group that you wish to use for the virtual machine.

- Use the `Extensions` pane to identify a script that contains the password that allows access to the Ark console. Create a file named `startup-password.sh` that contains the following text:

  ```
  #!/bin/sh
  rm -f /var/ppcd/startup-password.txt
  echo "console_password" > /var/ppcd/startup-password.txt
  chown ppcd:ppcd /var/ppcd/startup-password.txt
  chmod 600 /var/ppcd/startup-password.txt
  ```

  Where *`console_password`* is replaced with the password you will provide when prompted for a password by the Ark setup dialog.

To provide the location of the script to the virtual machine, open the `Extensions` pane, and click `Add extension`; when the `New resource` pane opens, select `Custom Script For Linux`.

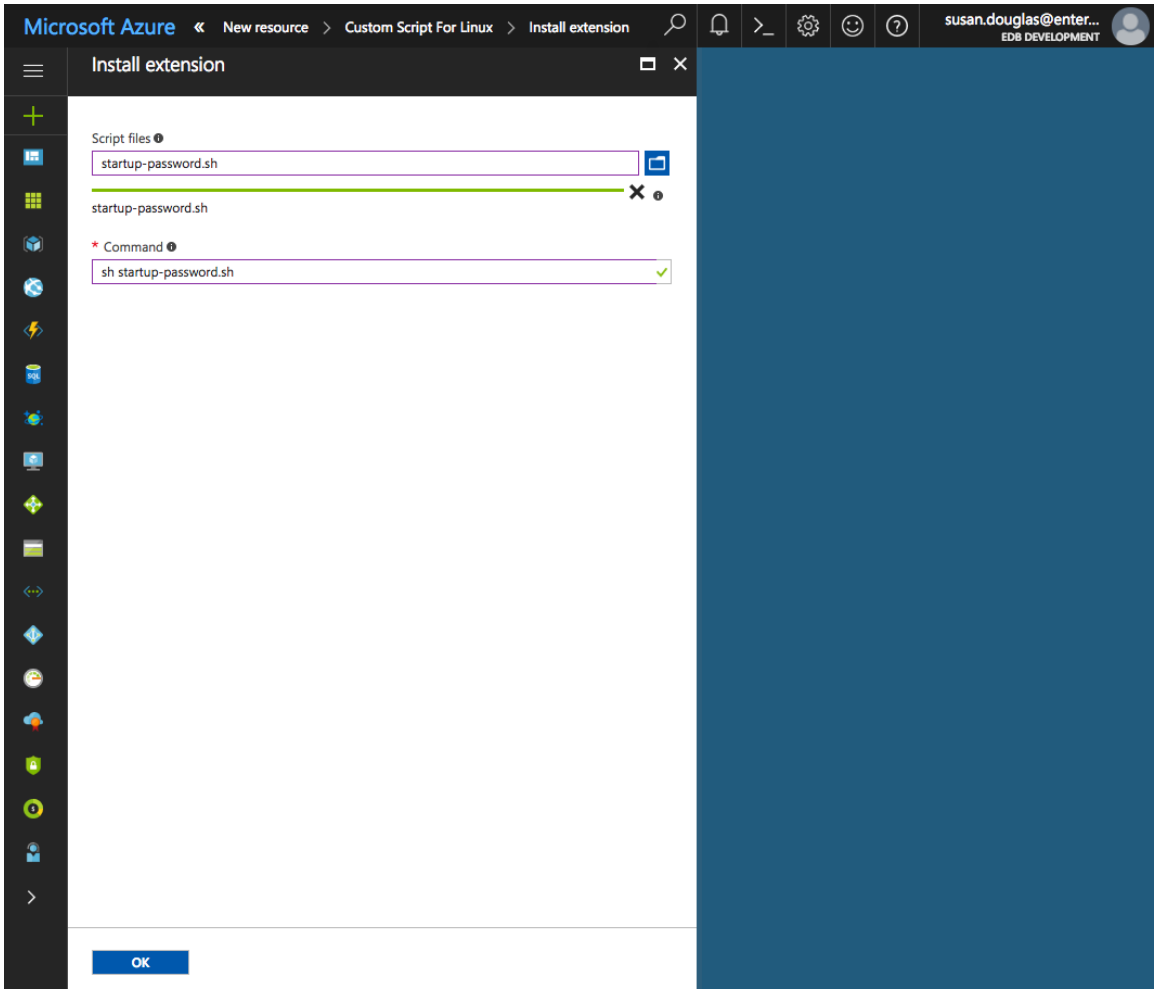

Figure 3.58 – Installing the extension.

Use the `Install extension` pane (see Figure 3.58) to identify the script file:

- Use the button to the right of the `Script files` field to open a file browser and upload the script file.
- Enter the command that will invoke your script in the `Command` field; for example, `sh startup-password.sh`.

Click `OK` to continue and return to the `Settings` panel; when you've finished updating the settings with your preferences, click `OK` to continue to the `Summary` panel.

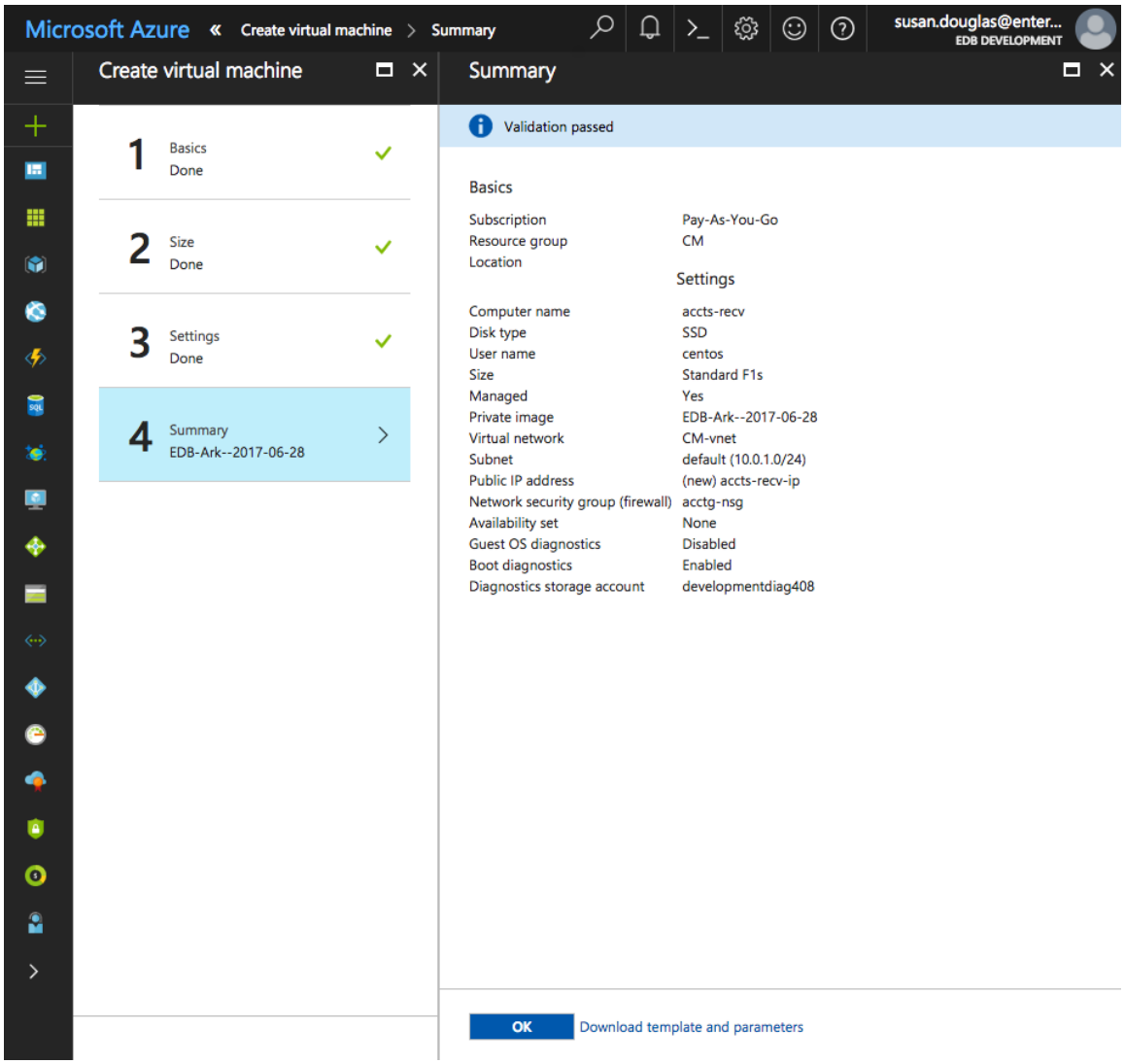

Figure 3.59 – The Azure Summary panel.

The `Summary` panel (see Figure 3.59) displays a detailed description of the configuration of the virtual machine that will host the Ark console. Select `OK` to begin deploying the virtual machine.

You can monitor the virtual machine's deployment from the Azure `Operations` page, the `Resource group` activity log, or the `Virtual machine` page. A notification will be generated when the deployment completes.

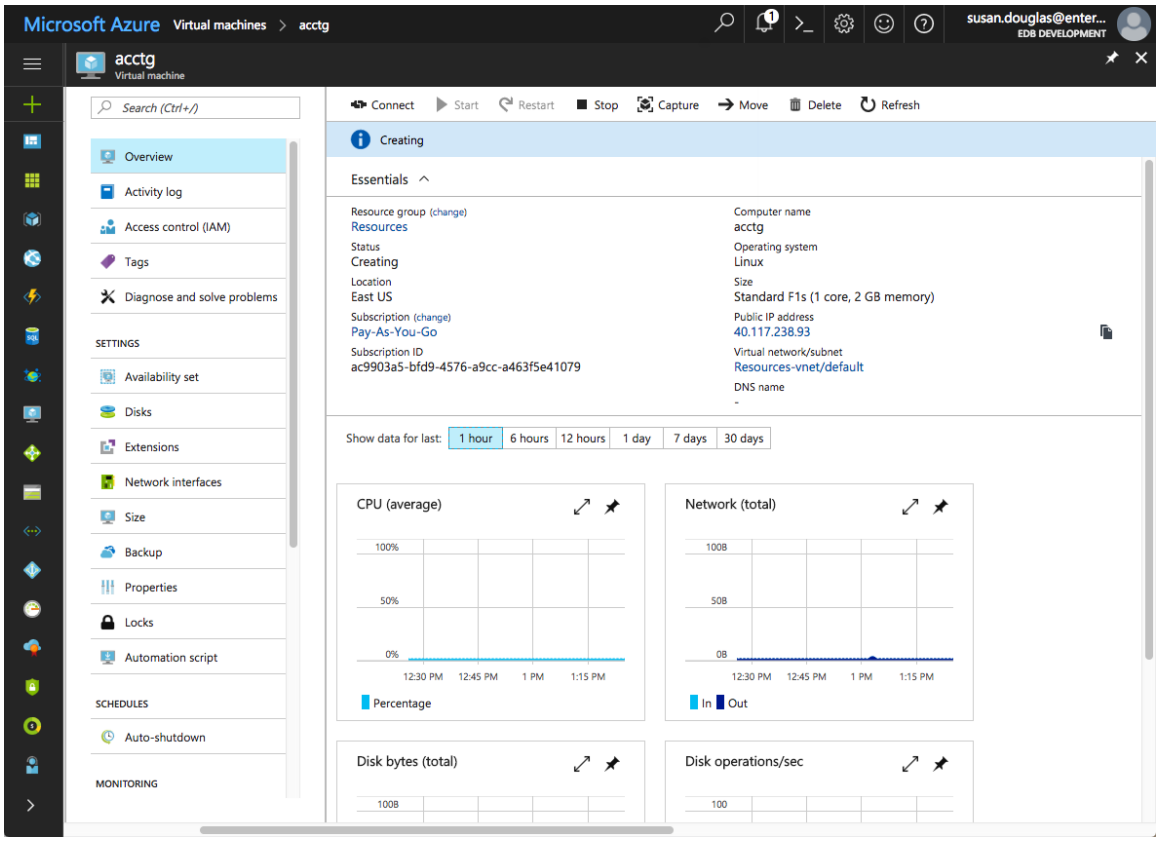

Figure 3.60 – The Virtual Machine details page.

While the virtual machine deploys, you can register your application in the Azure Active Directory. You will need the `Public IP address` or DNS name of your server for the registration. To copy the IP address, click the copy icon to the right of the `Public IP address` on the VM `Essentials` panel (see Figure 3.60).

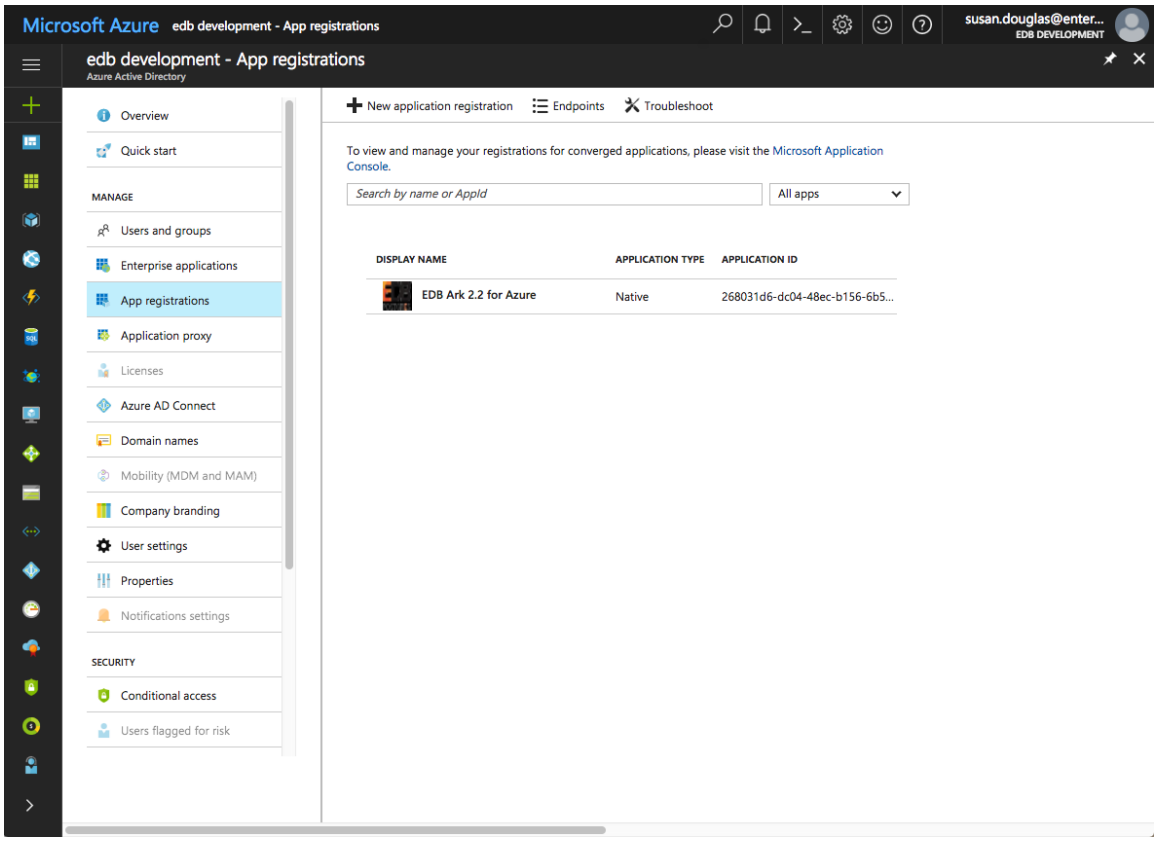

Figure 3.61 – The New application registration page.

After copying the public IP address or DNS name of your server, select `App registrations` from the `Active Directory` page. Click the `New application registration` button located on the `App registrations` detail panel (see Figure 3.61).

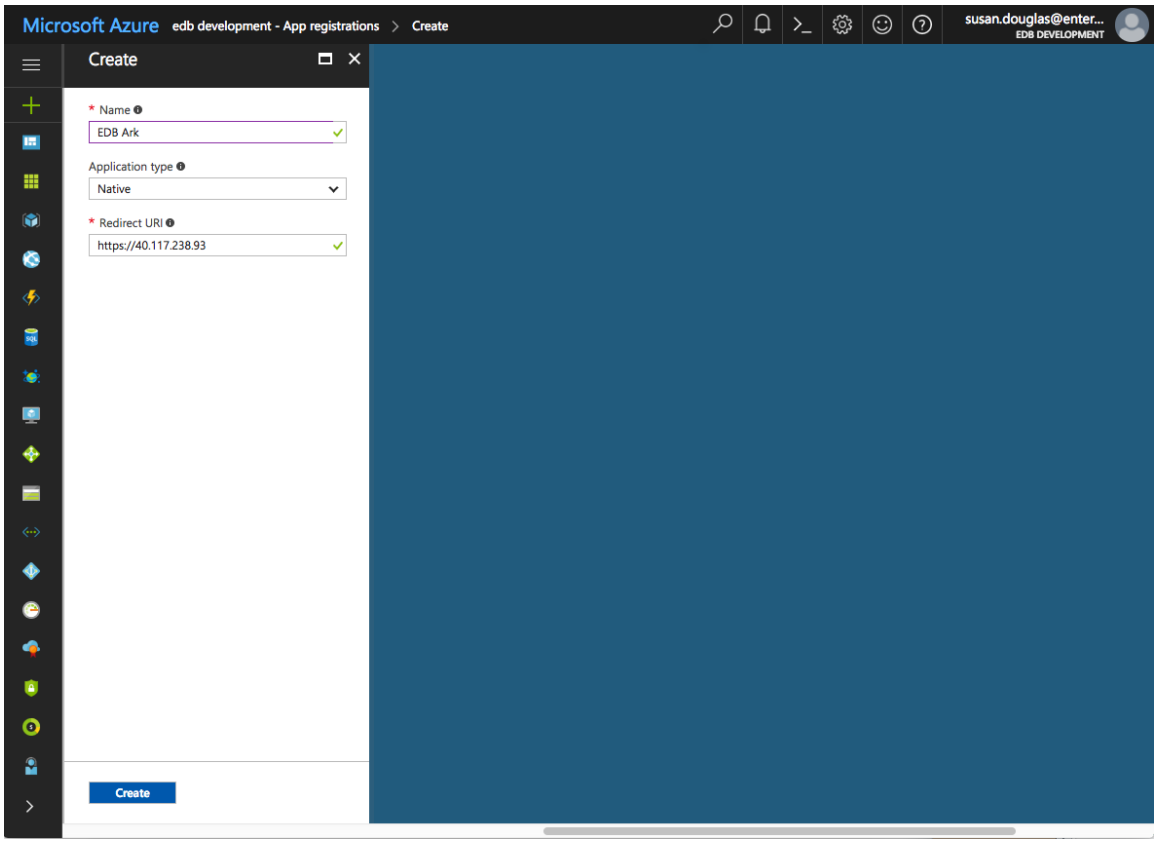

Figure 3.62 – The New application registration page.

Use fields on the `Create` panel (see Figure 3.62) to provide information about your application:

- Provide the application name in the Name field.
- Use the drop-down listbox in the Application type field to select the Application type; select Native for the Ark console application.
- Provide the public IP address of the virtual machine that is hosting the console in the Redirect URI field.

Click `Create` to register your application.

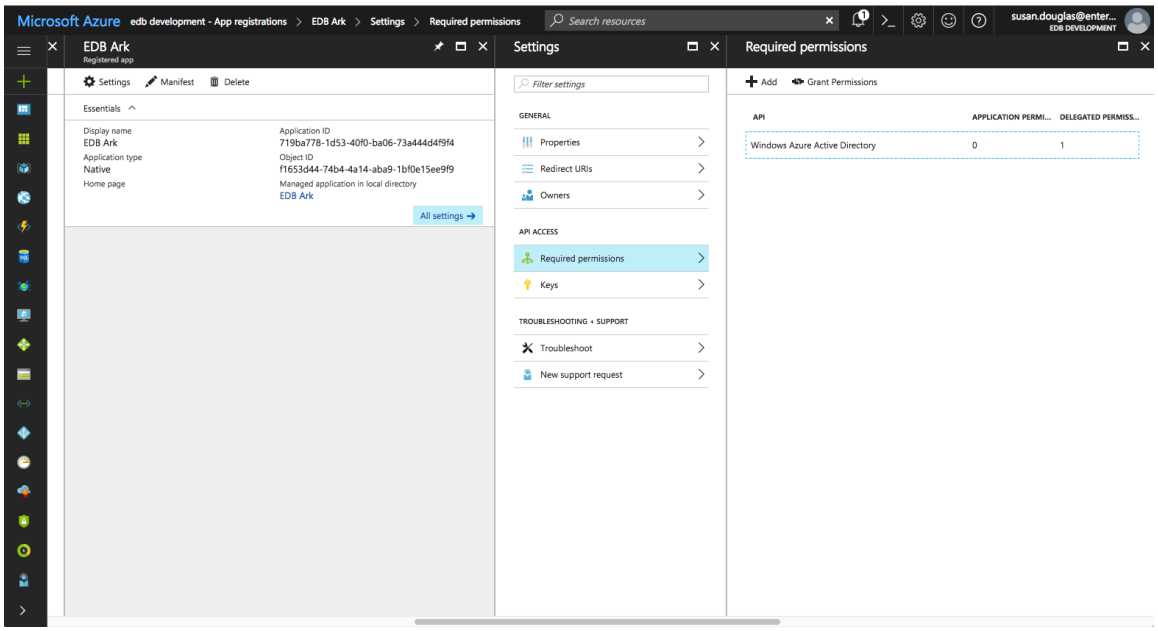

Figure 3.63 – The Required permissions page.

After creating the virtual machine and registering the application, you must adjust the required permissions, allowing the Windows Azure Service Management API to connect to your application. This will give the Ark server permission to control Azure services via the Service Management API.

Please note that you must be an Azure Global Administrator to grant permissions required by Ark. Navigate to the `Required permissions` page for the application, and select `+Add` from the `Required permissions` panel (see Figure 3.63).

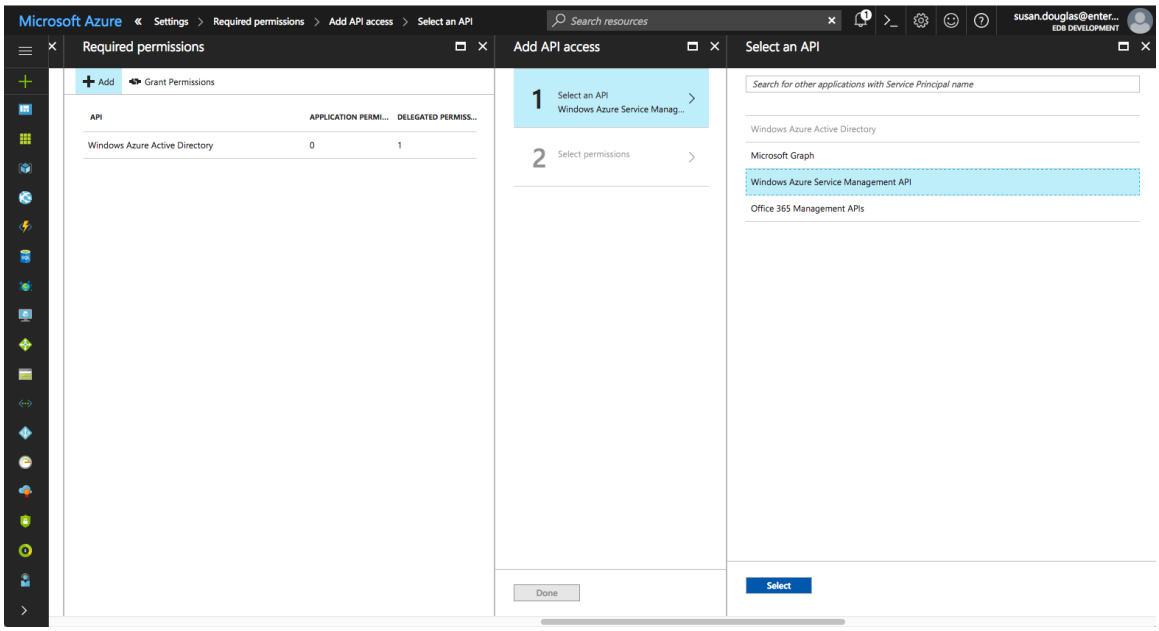

Figure 3.64 – Selecting an API.

Click `Select an API`, and then highlight `Windows Azure Service Management API` (see Figure 3.64).

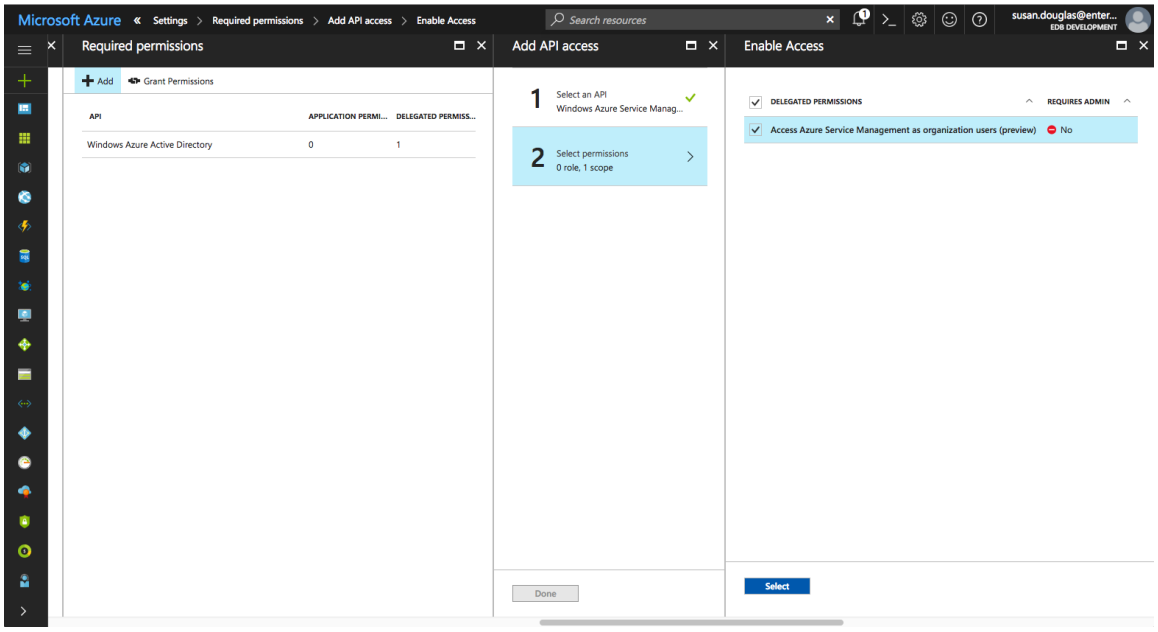

Figure 3.65 – Specifying API permissions.

Click `Select permissions`, and then `Access Azure Service Management` (see Figure 3.65); then, click `Select`.

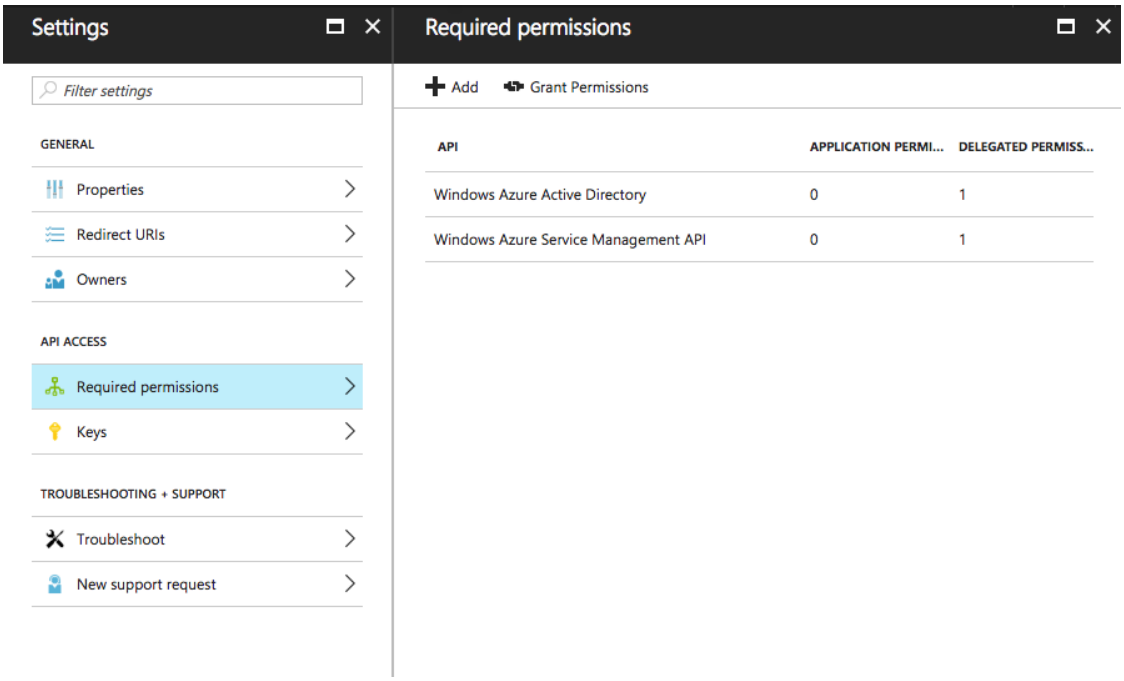

Figure 3.66 – Confirming that the permissions are added.

Then, click `Grant Permissions` (see Figure 3.66).

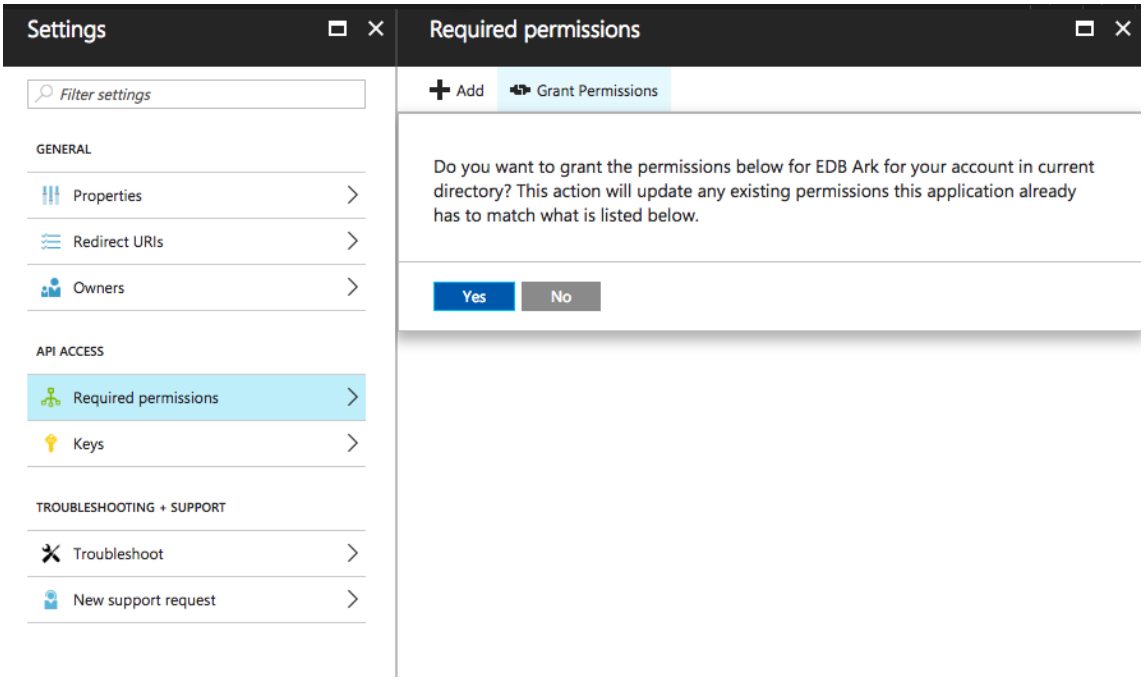

Figure 3.67 – Granting permissions for the Application.

When prompted, click `Yes` to confirm that you wish to grant access permissions (see Figure 3.67).

Repeat the process, adding permissions for `Microsoft Graph`. When adding permissions for `Microsoft Graph`, select a scope of `Read all users' full profiles`.

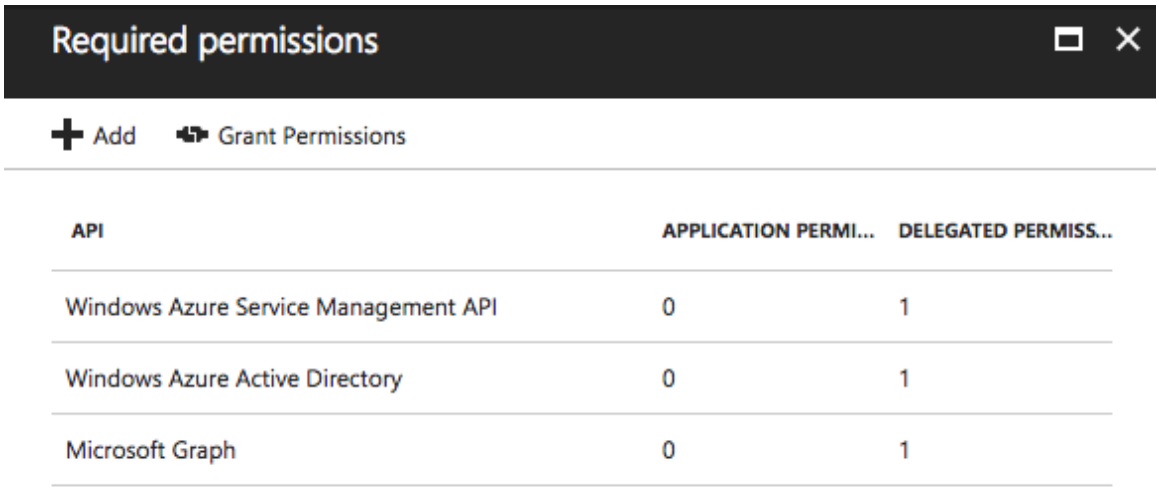

Figure 3.68 – Sufficient Resource permissions.

When you're finished granting permissions, the `Required permissions` list (see Figure 3.68) should include:

- Wizard Azure Active Directory
- Windows Azure Service Management API
- Microsoft Graph

3.3.5 Configuring the Ark Console

To configure the Ark console, you will be required to provide the password assigned to the Ark console when the console image was deployed. If you did not assign a password in a script identified as an extension (when creating the Azure virtual machine), a password will be created randomly, and stored in `/var/ppcd/startup-password.txt`.

To retrieve a system assigned password, ssh into the console host:

```
[ppcd@acctg ~]$ more /var/ppcd/startup-password.txt
h020zdigm95xxqmjonrs
```

To access the Ark setup dialog and configure the console, open a browser window and navigate to the IP address assigned to the console.

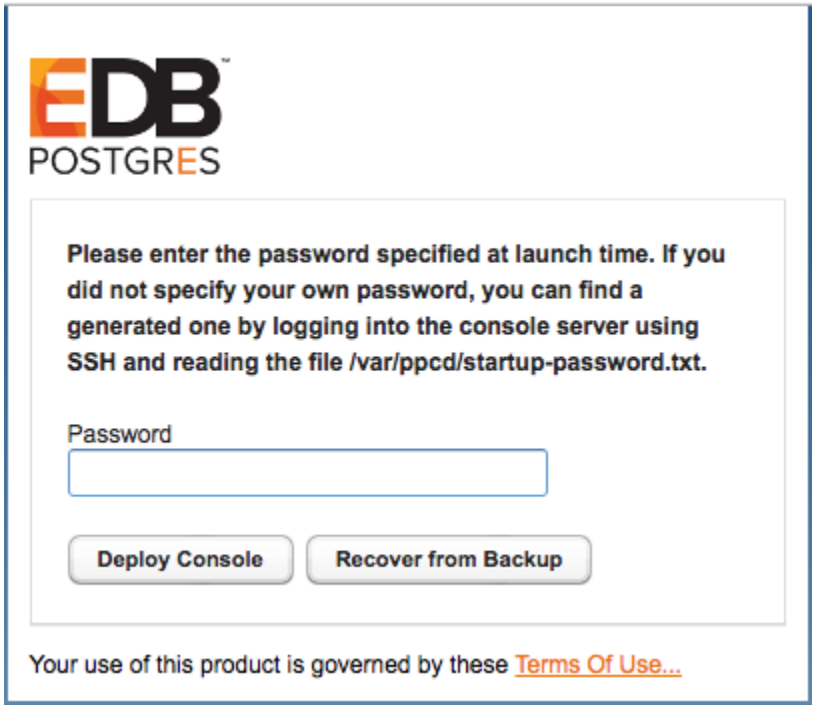

Figure 3.69 – Logging in to the instance.

When prompted, provide the console password (see Figure 3.69). The Ark console setup dialog opens as shown in Figure 3.70.

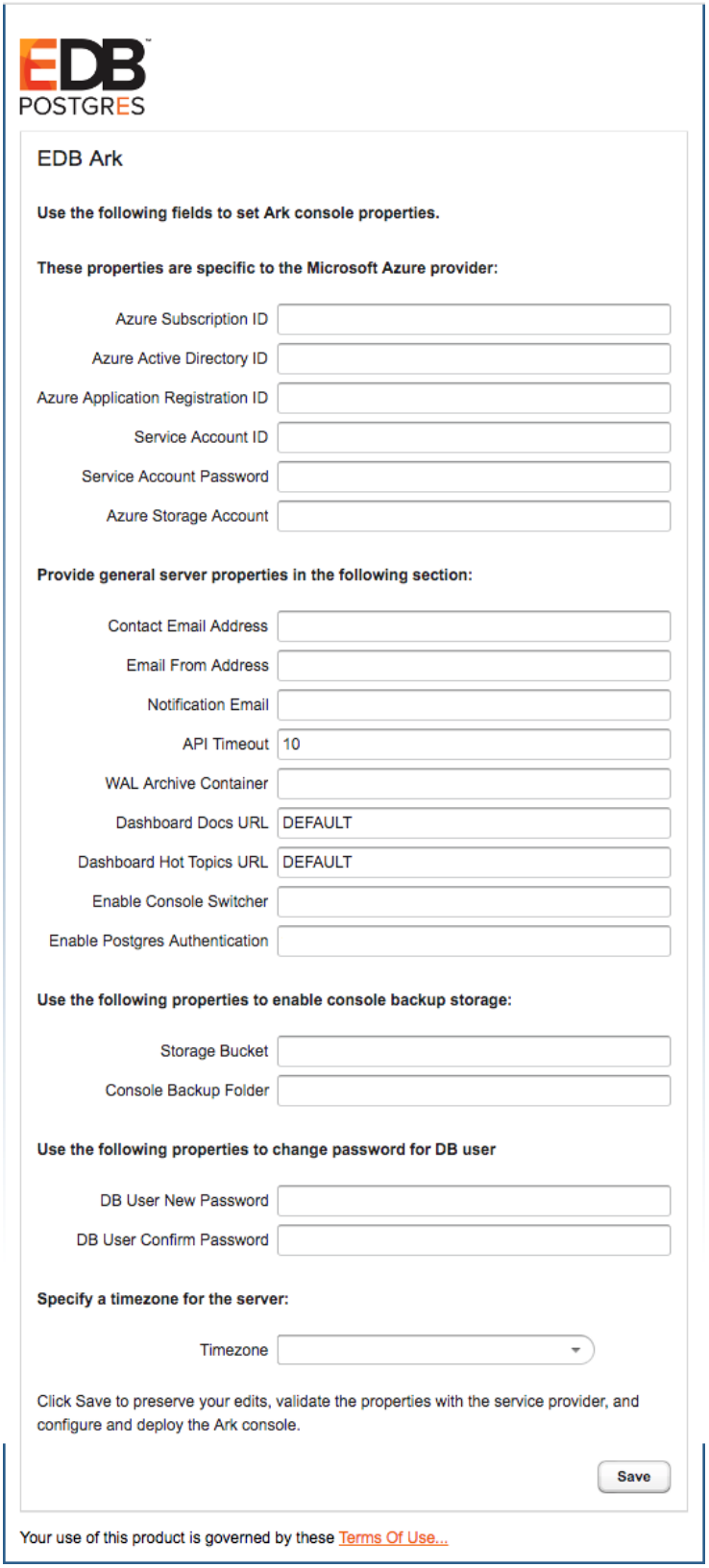

Figure 3.70 – Configuring the Ark console.

Use fields on the setup dialog to provide provider specific information and configuration details for the Ark console.

The fields in the first section of the setup dialog set values that are Azure specific:

- Use the Azure Subscription ID field to specify the subscription ID for the Azure account that hosts the Ark console. You can locate the subscription ID on the Azure Subscriptions page.

- Use the Azure Active Directory ID field to specify the directory ID associated with the Azure account that hosts the Ark console. To locate the directory ID, navigate to the Azure Active Directory and select Properties.

- Use the Azure Application Registration ID field to specify the application ID associated with the Azure account that hosts the Ark console. To locate the application ID, select Enterprise applications or App registrations from the Azure Active Directory menu; use the application ID associated with the registration created for the Ark console.

- Use the Service Account ID field to specify the name of the Azure service account. The service account must be an owner of the resource group in which the Ark server is deployed.

- Use the Service Account Password field to specify the password associated with the service account.

- Use the Azure Storage Account field to specify the name of the Azure block storage account you wish to use with this Ark server.

The fields in the General properties section set values that control Ark behaviors:

- Use the Contact Email Address field to specify the address that will be included in the body of cluster status notification emails.

- Use the Email From Address field to specify the return email address specified on cluster status notification emails.

- Use the Notification Email field to specify the email address to which email notifications about the status of the Ark console will be sent.

- Use the API Timeout field to specify the number of minutes that an authorization token will be valid for use within the API.

- Use the `WAL Archive Container` field to specify the name of the storage container where WAL archives (used for point-in-time recovery) are stored. You must provide a value for this property; once set, this property must not be modified.

- Use the `Dashboard Docs URL` field to specify the location 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 parameter to `DEFAULT` to display content (documentation) from EnterpriseDB; to display alternate content, provide the URL of the content. To display no content in the lower half of the `Dashboard` tab, leave the field blank.

- Use the `Dashboard Hot Topics URL` field to specify the location 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 parameter to `DEFAULT` to display content (alerts) from EnterpriseDB; to display alternate content, provide the URL of the content. To display no content across the middle section of the `Dashboard` tab, leave the field blank.

- Use the `Enable Console Switcher` field to indicate if the console should display console switcher functionality. When set to `true`, the console will display the switcher; when `false`, the switcher will not be displayed. For more information, see Section 4.1.1.

- Set `Enable Postgres Authentication` to `true` to instruct Ark to enforce the authentication method configured on the backing Postgres server. Supported authentication methods include password, LDAP, RADIUS, PAM, and BSD.

  If `false`, Ark will use the default authentication method (password).

Use fields in the next section to provide information about the location of the console backup storage in the next section of the setup dialog; please note that you must provide values in these fields to use the Ark console recovery functionality:

- Use the `Storage Bucket` field to specify the name of the container that will be used to store files for point-in-time recovery. This location may not change after the initial deployment of the Ark console.

  - A container name must be at least 3 and no more than 63 characters in length.

  - A container name may contain lowercase letters, numbers, and the dash character (-).

  - A container name must start with a lowercase letter or number.

For more information, please see the Azure documentation at:

https://docs.microsoft.com/en-us/rest/api/storageservices/naming-and-referencing-containers--blobs--and-metadata

- Use the `Console Backup Folder` field to specify a folder in which the backups will be stored.

Use the last field to specify a timezone for the server:

- Use the drop-down listbox in the `Timezone` field to select the timezone that will be displayed by the Ark console.

When you've completed the setup dialog, click the `Save` button to validate your changes and restart the server.

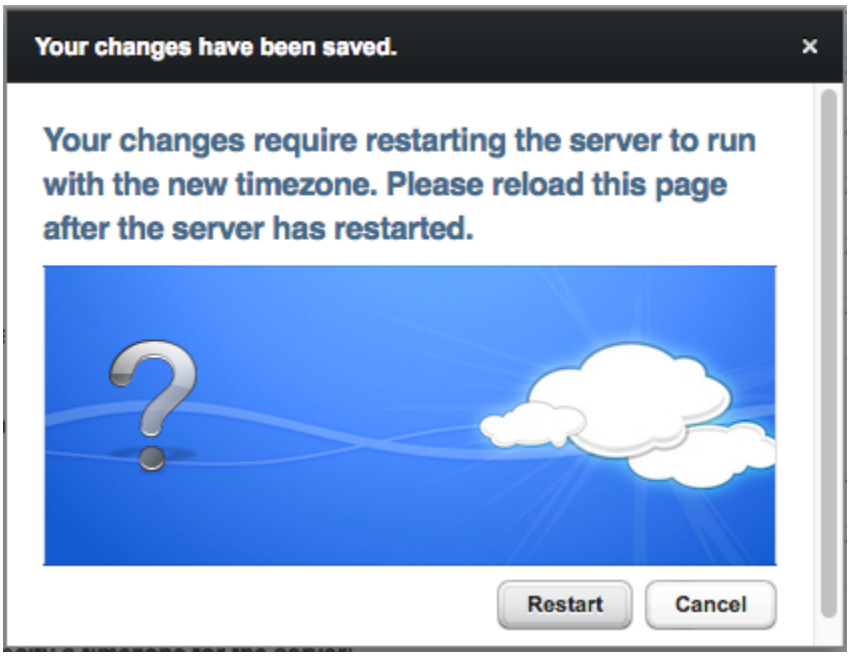

Figure 3.71 – Restart the server to start the Ark console.

When prompted, click the `Restart` button to restart the server and start the Ark console (see Figure 3.71). Ark will confirm that the server is restarting (see Figure 3.72).

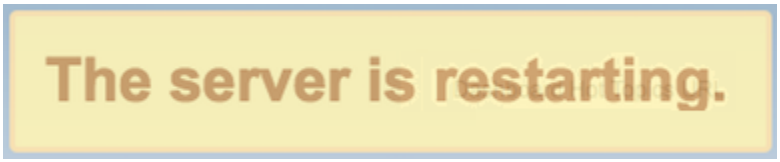

Figure 3.72 – The server restart message.

3.3.6 Connecting to the Administrative Console on an Azure Host

When you navigate to the URL of the installed Ark console that uses Azure to host clusters, the console will display a login dialog (see Figure 3.73).

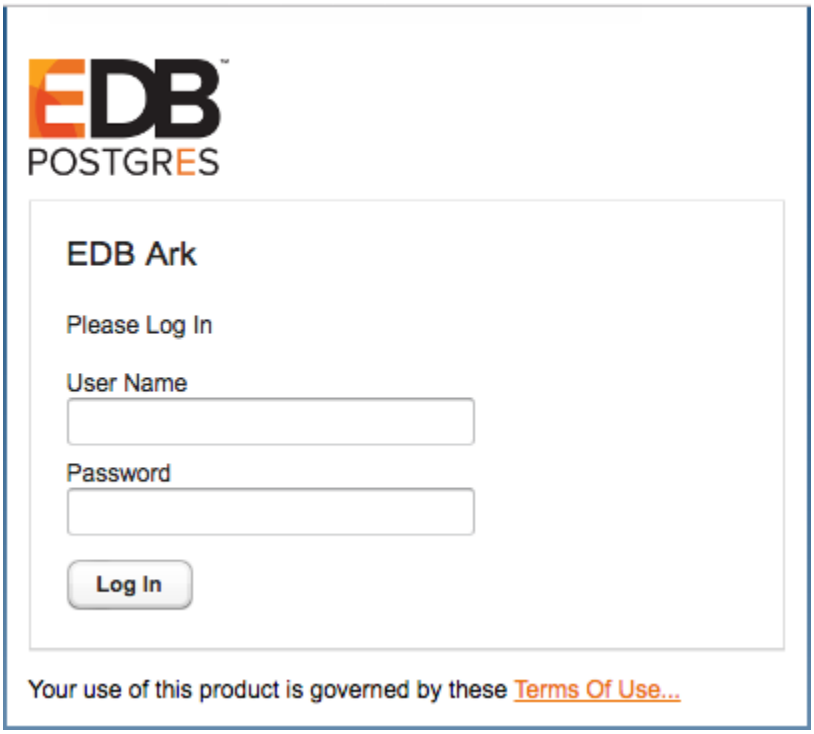

Figure 3.73 - The Login dialog.

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 a role with administrative privileges, the Ark console will include the `DBA` tab and the `Admin` tab (as shown in Figure 3.74).

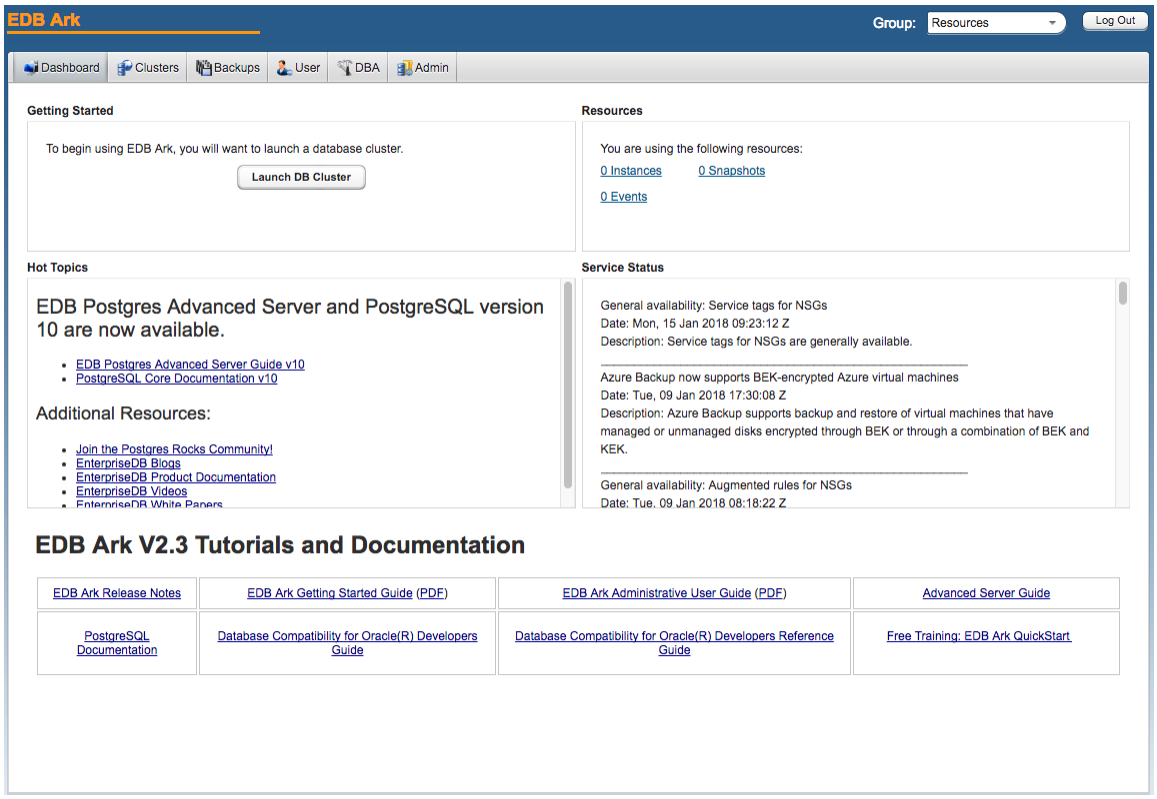

Figure 3.74 - The Dashboard of the EDB Ark Administrator's console.

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.

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).

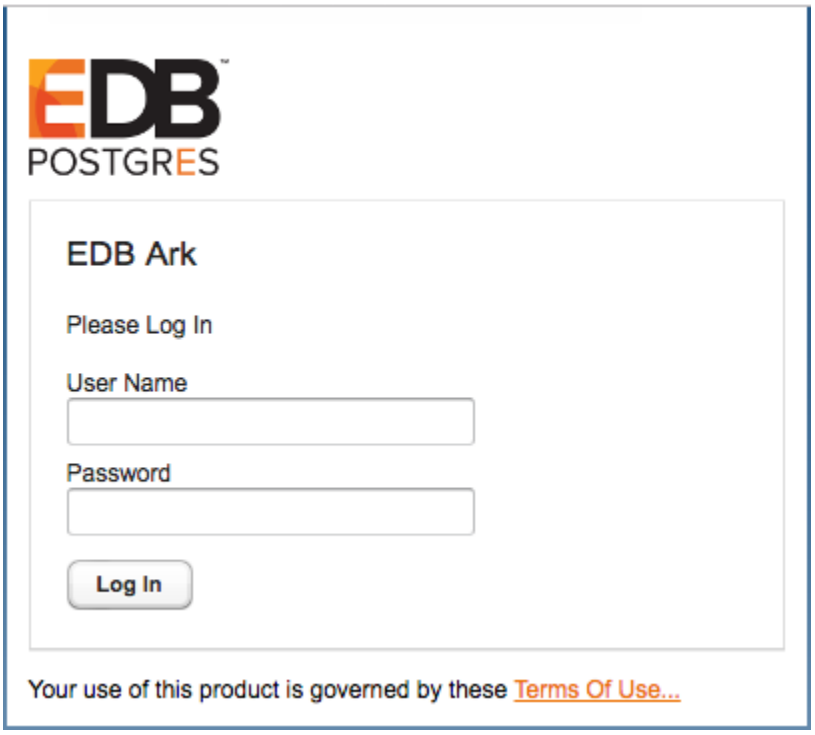

Figure 4.1 - The Login dialog.

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.

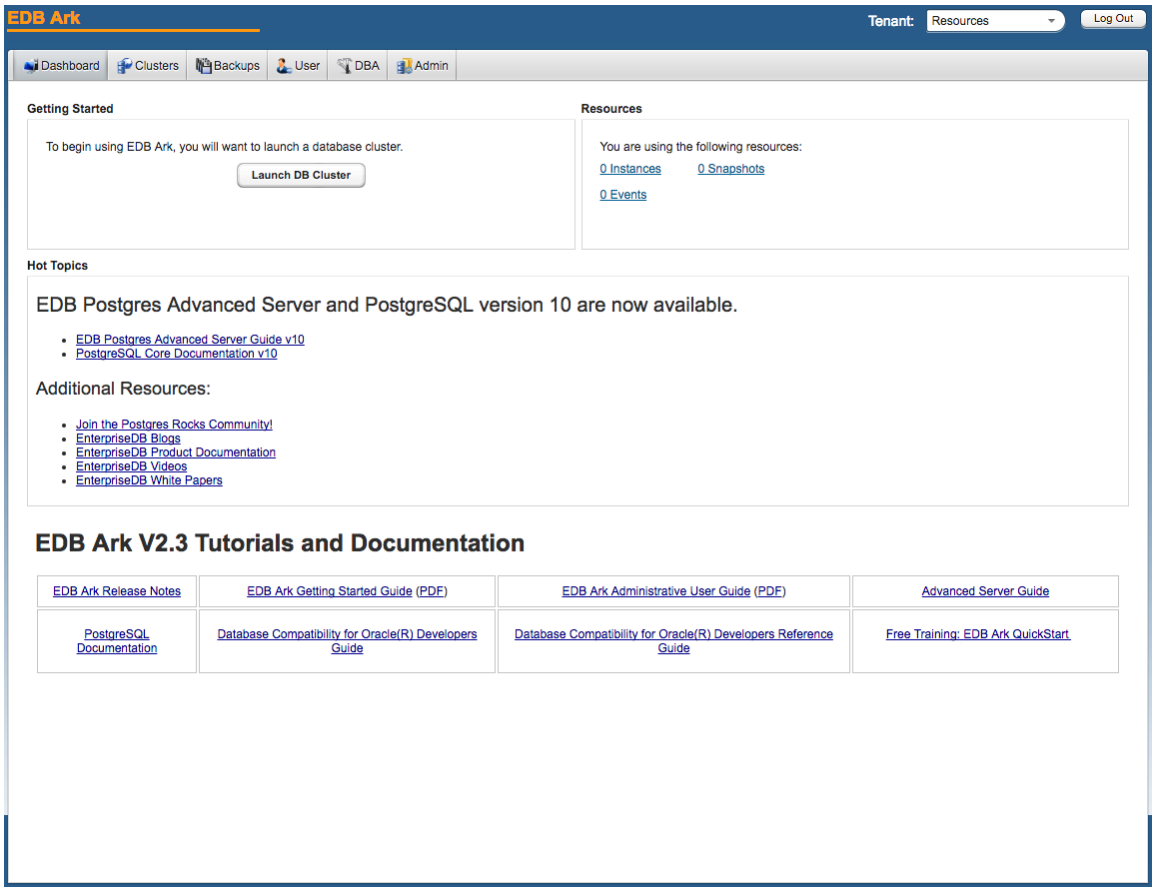

Figure 4.2 - The EDB Ark Administrator's console.

4.1 Using the Admin Tab

Use options on the `Admin` tab (see Figure 4.3) to perform platform-specific administrative tasks.

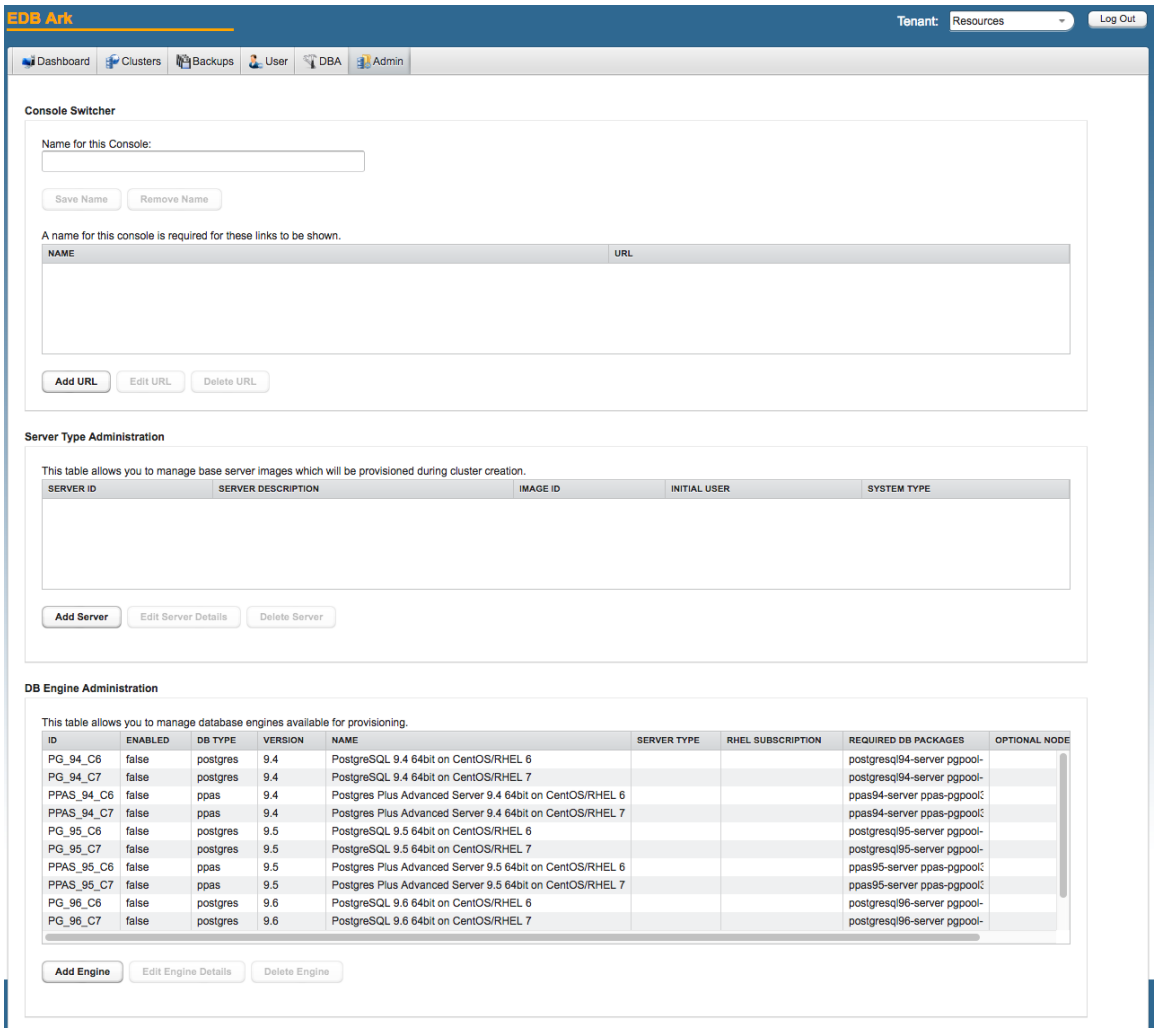

Figure 4.3 – The Admin tab

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, 9.5, and 9.6 (64-bit)
- PostgreSQL 9.4, 9.5, and 9.6 (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.

RHEL Subscription Management

Options in the `RHEL Subscription Management` box allow you to:

- Add, modify, or delete RHEL subscription information. For more information, see Section 4.1.4.

IAM Roles Administration (AWS only)

Options in the `IAM Roles Administration` box allow you to:

- Make Amazon ARNs available for use in Ark user accounts (AWS). For information about user administration options, see Section 4.1.5.

User Administration

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

- If applicable, manage user accounts.
- Access a list of currently connected users.
- Display a banner message to connected users.

For information about user administration options, see Section 4.1.6.

Download Console Logs

Click 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 4.1.7.

Edit Installation Properties

Click the `Edit installation properties` button to open a dialog that allows you to modify the Ark console configuration. For more information, see Section 4.1.8.

4.1.1 Using the Console Switcher

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.

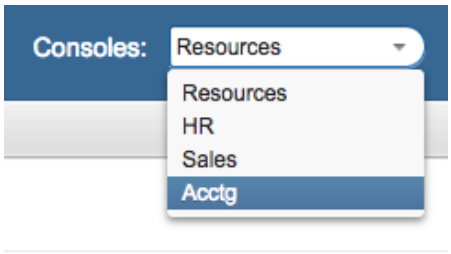

Figure 4.4 – The Consoles drop-down.

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).

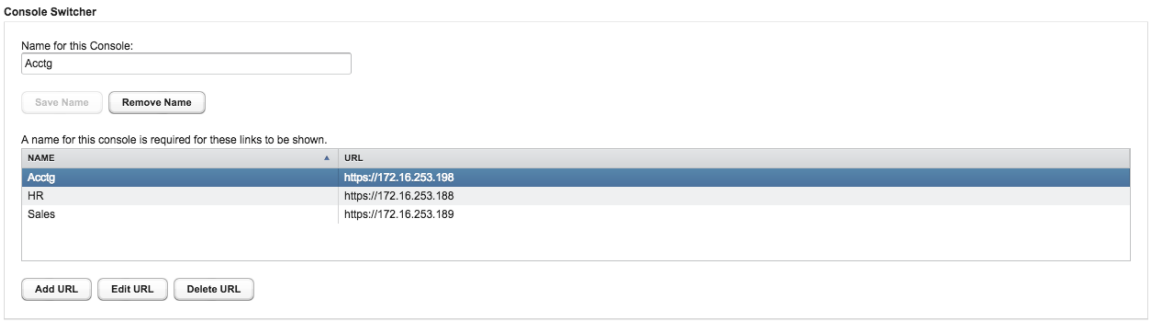

Figure 4.5 – The Console Switcher section of the Admin tab.

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.

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).

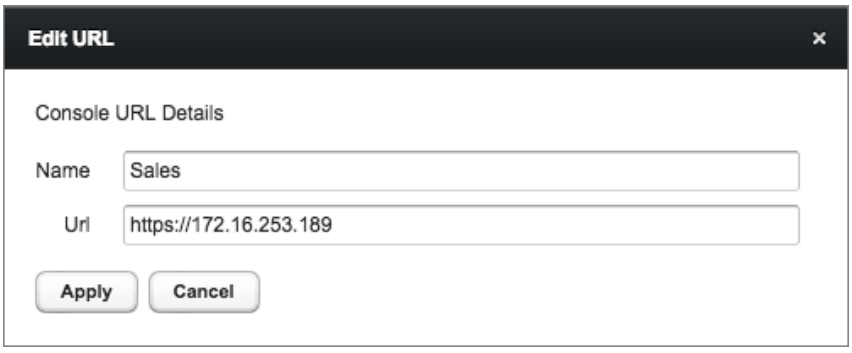

Figure 4.8 – The Edit URL dialog.

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).

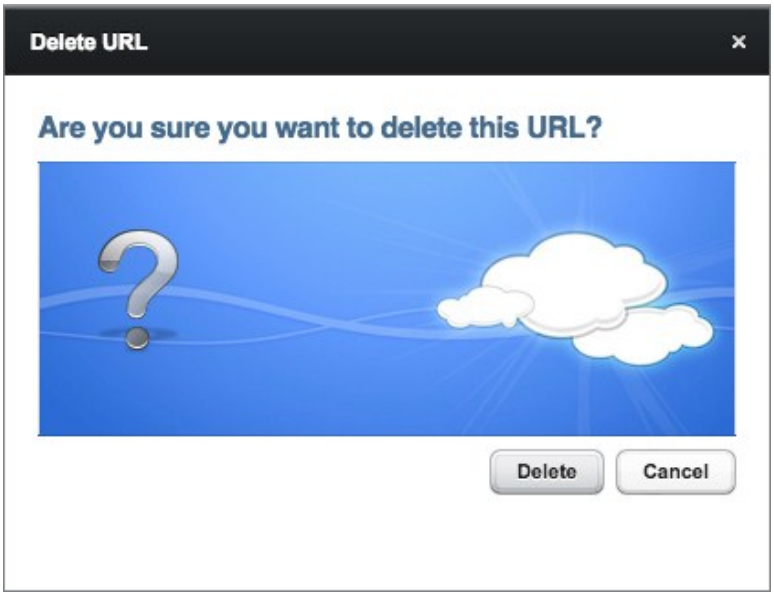

Figure 4.9 – The Edit URL dialog.

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.

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).

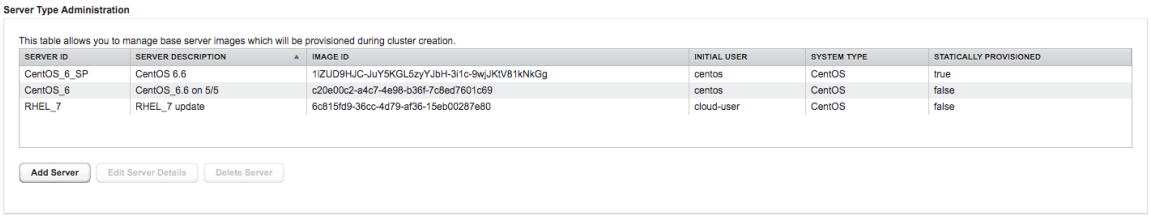

Figure 4.10 – The Server Type Administration section of the Admin tab.

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 `Add Server` dialog (shown in Figure 4.11) opens.

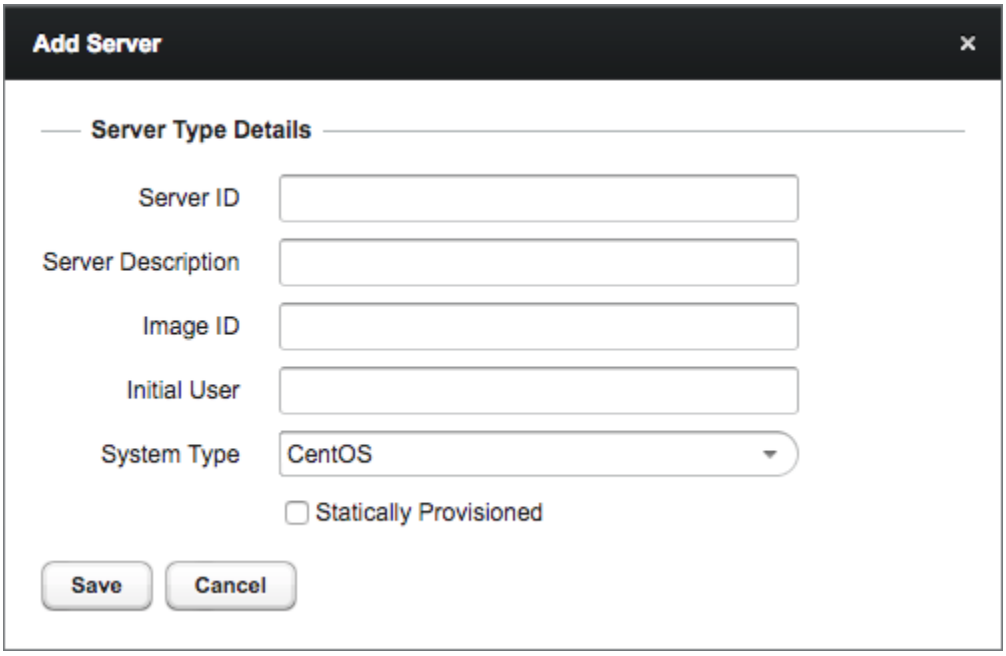

Figure 4.11 – The Add Server dialog.

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.

- 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

  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.

  On Amazon

  If you are using Ark with Amazon, provide the AMI ID in the Image ID field. Please note: you should use a server from a trusted source, with a virtualization type of hvm. We recommend using the official Amazon images from the Amazon AWS Marketplace.

  On Azure

  If you are using an Ark with Azure, you can use the Azure CLI interface to retrieve a list of the machine images that are available in the Azure Marketplace. For information about downloading and installing the Azure CLI, visit:

  https://docs.microsoft.com/en-us/cli/azure/install-azure-cli

  After installing the Azure CLI, you can use one of the following commands to retrieve a list of the available images for a specific platform and version:

Version	Command
RHEL 7:	`az vm image list --offer RHEL --sku 7. --output table --all`
CentOS 7:	`az vm image list --offer CentOS --sku 7. --output table --all`
CentOS 6:	`az vm image list --offer CentOS --sku 6. --output table --all`

  Select an image from a trusted Publisher; when configuring the Ark console, provide the first three elements of the Urn column in the Server Image ID field. For example, if the Urn returned by the CLI is RedHat:RHEL:7.2:7.2.20160921 7.2.20160921, the Image ID is RedHat:RHEL:7.2.

Some recommended images and providers are:

- RedHat:RHEL:7.4
- OpenLogic:CentOS:7.4
- OpenLogic:CentOS:6.9

- Use the `Initial User` field to provide the name of the default operating system user. This user must have `sudo root` privileges to perform the initial provisioning of software on the node.

  If you are using an Amazon AWS Marketplace image, the default user name is associated with the backing image; for more information about image user identities, see:

  https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/AccessingInstancesLinux.html

- Use the `System Type` field drop-down listbox to select the operating system type of the server; select `CentOS` or `RHEL`.

- Check the box next to `Statically Provisioned` to indicate that the server is statically provisioned. A statically provisioned server is a pre-installed image that contains the software required to create a database cluster.

  For detailed information about creating a statically provisioned image, please see Section 10.5.

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.

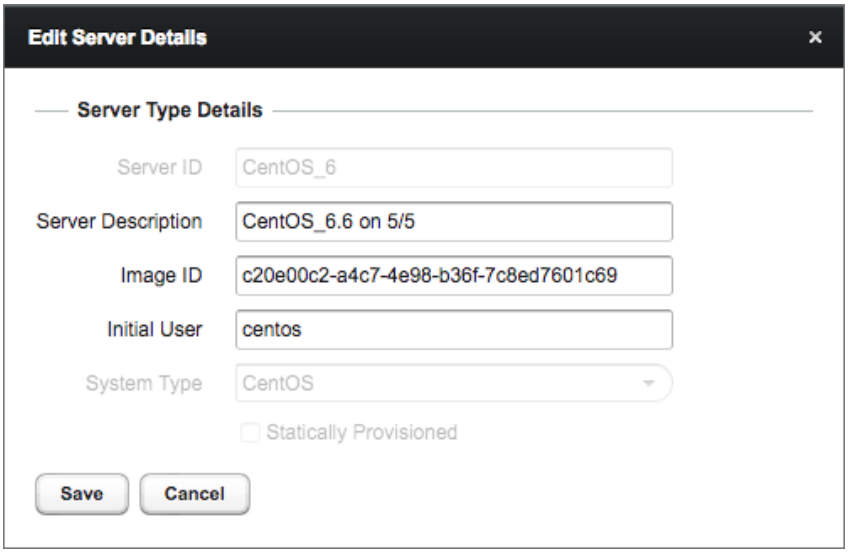

Figure 4.12 – The Edit Server dialog.

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

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).

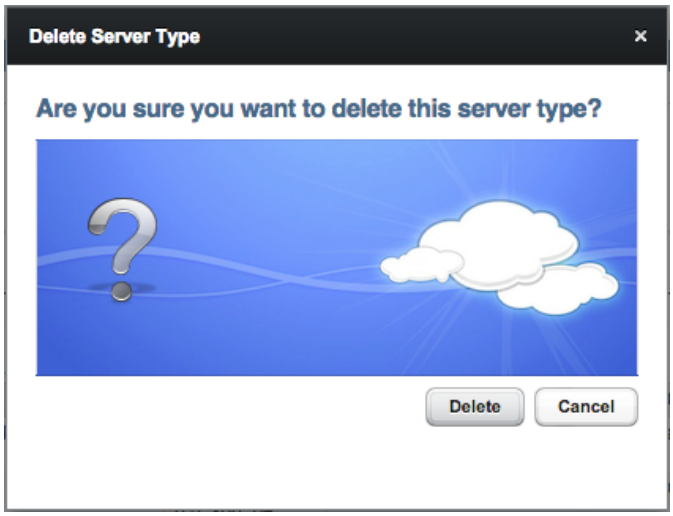

Figure 4.13 – The Delete Server Type dialog.

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.

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).

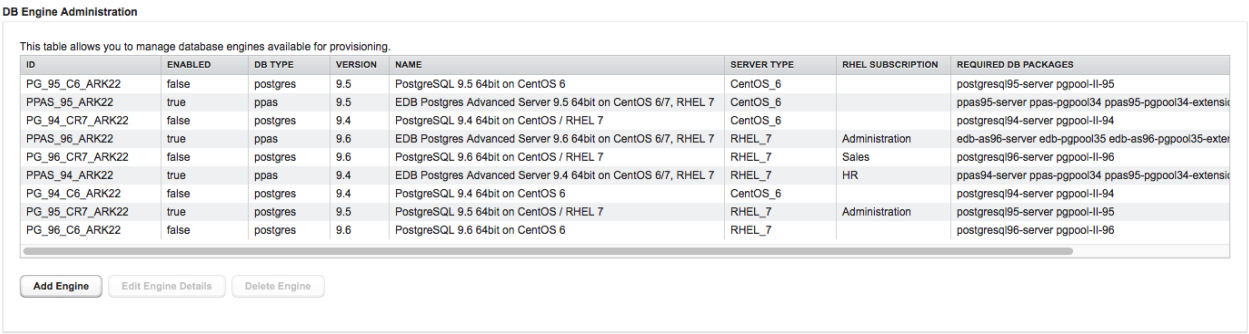

Figure 4.15 – The DB Engine Administration section of the Admin tab.

The Ark console ships with a number of default engine definitions. Before using an engine, you must create servers (see Section 4.1.2) and edit the engine details, associating a server with the engine you wish to use and enabling the engine.

The following engines are shipped with Ark. Please note that the engine definitions include multiple repositories to provide access to all of the packages required to complete the installation. Advanced Server repositories require you to provide a `USERNAME` and associated `PASSWORD`; to request a username and password, visit

https://www.enterprisedb.com/general-inquiry-form

PostgreSQL 9.3 64bit on CentOS 6

Repository Locations:
`http://yum.postgresql.org/9.3/redhat/rhel-6-x86_64/pgdg-centos93-9.3-3.noarch.rpm`
`http://USERNAME:PASSWORD@yum.enterprisedb.com/tools/redhat/rhel-$releasever-$basearch`
`http://USERNAME:PASSWORD@yum.enterprisedb.com/dependencies/redhat/rhel-$releasever-$basearch`

Required Packages: `postgresql93-server pgpool-II-93 pem-agent`

PostgreSQL 9.3 64bit on CentOS / RHEL 7

Repository Locations:
`https://yum.postgresql.org/9.3/redhat/rhel-7-x86_64/pgdg-redhat93-9.3-3.noarch.rpm`
`http://USERNAME:PASSWORD@yum.enterprisedb.com/tools/redhat/rhel-$releasever-$basearch`
`http://USERNAME:PASSWORD@yum.enterprisedb.com/dependencies/redhat/rhel-$releasever-$basearch`

Required Packages: `postgresql93-server pgpool-II-93 pem-agent`

PostgreSQL 9.4 64bit on CentOS 6

Repository Location:
`http://yum.postgresql.org/9.4/redhat/rhel-6-x86_64/pgdg-centos94-9.4-3.noarch.rpm`
`http://USERNAME:PASSWORD@yum.enterprisedb.com/tools/redhat/rhel-$releasever-$basearch`
`http://USERNAME:PASSWORD@yum.enterprisedb.com/dependencies/redhat/rhel-$releasever-$basearch`

Required Packages: `postgresql94-server pgpool-II-94 pem-agent`

PostgreSQL 9.4 64bit on CentOS / RHEL 7

Repository Location:
`http://yum.postgresql.org/9.4/redhat/rhel-7-x86_64/pgdg-redhat94-9.4-3.noarch.rpm`
`http://USERNAME:PASSWORD@yum.enterprisedb.com/tools/redhat/rhel-$releasever-$basearch`
`http://USERNAME:PASSWORD@yum.enterprisedb.com/dependencies/redhat/rhel-$releasever-$basearch`

Required Packages: `postgresql94-server pgpool-II-94 pem-agent`

EDB Postgres Advanced Server 9.4 64bit on CentOS 6/7, RHEL 7

Repository Locations:
`http://USERNAME:PASSWORD@yum.enterprisedb.com/9.4/redhat/rhel-$releasever-$basearch`
`http://USERNAME:PASSWORD@yum.enterprisedb.com/tools/redhat/rhel-$releasever-$basearch`
`http://USERNAME:PASSWORD@yum.enterprisedb.com/dependencies/redhat/rhel-$releasever-$basearch`

Required Packages: `ppas94-server ppas-pgpool34 ppas94-pgpool34-extensions pem-agent`

PostgreSQL 9.5 64bit on CentOS 6

Repository Location:
`http://yum.postgresql.org/9.5/redhat/rhel-6-x86_64/pgdg-centos95-9.5-3.noarch.rpm`
`http://USERNAME:PASSWORD@yum.enterprisedb.com/tools/redhat/rhel-$releasever-$basearch`
`http://USERNAME:PASSWORD@yum.enterprisedb.com/dependencies/redhat/rhel-$releasever-$basearch`

Required Packages: `postgresql95-server pgpool-II-95 pem-agent`

PostgreSQL 9.5 64bit on CentOS / RHEL 7

Repository Location:
`http://yum.postgresql.org/9.5/redhat/rhel-7-x86_64/pgdg-redhat95-9.5-3.noarch.rpm`
`http://USERNAME:PASSWORD@yum.enterprisedb.com/tools/redhat/rhel-$releasever-$basearch`
`http://USERNAME:PASSWORD@yum.enterprisedb.com/dependencies/redhat/rhel-$releasever-$basearch`

Required Packages: `postgresql95-server pgpool-II-95 pem-agent`

EDB Postgres Advanced Server 9.5 64bit on CentOS 6/7, RHEL 7

Repository Locations:
`http://USERNAME:PASSWORD@yum.enterprisedb.com/9.5/redhat/rhel-$releasever-$basearch`
`http://USERNAME:PASSWORD@yum.enterprisedb.com/tools/redhat/rhel-$releasever-$basearch`
`http://USERNAME:PASSWORD@yum.enterprisedb.com/dependencies/redhat/rhel-$releasever-$basearch`

Required Packages: `ppas95-server ppas-pgpool34 ppas95-pgpool34-extensions pem-agent`

PostgreSQL 9.6 64bit on CentOS 6

Repository Location:
`http://yum.postgresql.org/9.6/redhat/rhel-6-x86_64/pgdg-centos96-9.6-3.noarch.rpm`
`http://USERNAME:PASSWORD@yum.enterprisedb.com/tools/redhat/rhel-$releasever-$basearch`
`http://USERNAME:PASSWORD@yum.enterprisedb.com/dependencies/redhat/rhel-$releasever-$basearch`

Required Packages: `postgresql96-server pgpool-II-96 pem-agent`

PostgreSQL 9.6 64bit on CentOS / RHEL 7

Repository Location:
`http://yum.postgresql.org/9.6/redhat/rhel-7-x86_64/pgdg-redhat96-9.6-3.noarch.rpm`
`http://USERNAME:PASSWORD@yum.enterprisedb.com/tools/redhat/rhel-$releasever-$basearch`
`http://USERNAME:PASSWORD@yum.enterprisedb.com/dependencies/redhat/rhel-$releasever-$basearch`

Required Packages: `postgresql96-server pgpool-II-96 pem-agent`

EDB Postgres Advanced Server 9.6 64bit on CentOS 6/7, RHEL 7

Repository Locations:
`http://USERNAME:PASSWORD@yum.enterprisedb.com/9.6/redhat/rhel-$releasever-$basearch`
`http://USERNAME:PASSWORD@yum.enterprisedb.com/tools/redhat/rhel-$releasever-$basearch`
`http://USERNAME:PASSWORD@yum.enterprisedb.com/dependencies/redhat/rhel-$releasever-$basearch`

Required Packages: `edb-as96-server edb-pgpool35 edb-as96-pgpool35-extensions pem-agent`

PostgreSQL 10 64bit on CentOS 6

Repository Locations:
`https://yum.postgresql.org/10/redhat/rhel-6-x86_64/pgdg-redhat10-10-2.noarch.rpm`
`http://USERNAME:PASSWORD@yum.enterprisedb.com/tools/redhat/rhel-$releasever-$basearch`
`http://USERNAME:PASSWORD@yum.enterprisedb.com/dependencies/redhat/rhel-$releasever-$basearch`

Required Packages: `postgresql10-server pgpool-II-10 pgpool-II-10-extensions pem-agent`

PostgreSQL 10 64bit on CentOS / RHEL 7

Repository Locations:
`https://yum.postgresql.org/10/redhat/rhel-7-x86_64/pgdg-redhat10-10-2.noarch.rpm`
`http://USERNAME:PASSWORD@yum.enterprisedb.com/tools/redhat/rhel-$releasever-$basearch`
`http://USERNAME:PASSWORD@yum.enterprisedb.com/dependencies/redhat/rhel-$releasever-$basearch`

Required Packages: `postgresql10-server pgpool-II-10 pgpool-II-10-extensions pem-agent`

EDB Postgres Advanced Server 10 64bit on CentOS 6/7, RHEL 7

Repository Locations:
`http://USERNAME:PASSWORD@yum.enterprisedb.com/10/redhat/rhel-$releasever-$basearch`
`http://USERNAME:PASSWORD@yum.enterprisedb.com/tools/redhat/rhel-$releasever-$basearch`
`http://USERNAME:PASSWORD@yum.enterprisedb.com/dependencies/redhat/rhel-$releasever-$basearch`

Required Packages: `edb-as10-server edb-pgpool36 edb-as10-pgpool36-extensions pem-agent`

4.1.3.1 Adding, Modifying, or Deleting Engine Definitions

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`.

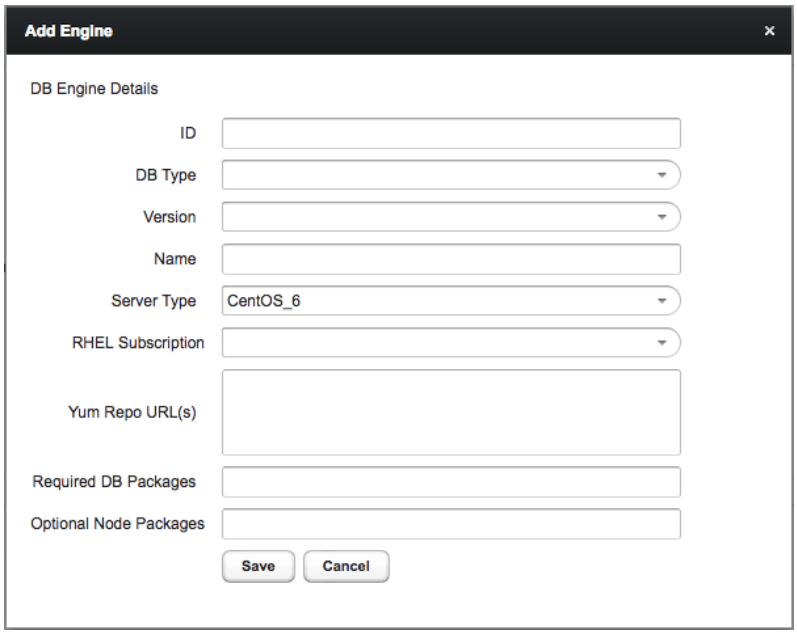

Figure 4.16 – The Add Engine dialog.

Use the fields on the `Add Engine` dialog to define a new server image/database pairing; please note that some fields are disabled if the server is statically provisioned:

- 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 drop-down listbox in the `RHEL Subscription` field to select the Red Hat Subscription Manager service that will be used by the engine. To populate the `RHEL Subscription` drop-down, describe your subscription services in the `RHEL Subscription Management` section of the `Admin` tab. RHEL Subscription Manager services are only applicable for RHEL 7 clusters.

  Please note that you must delete any instances that use an engine that is associated with a RHEL subscription before you can delete the RHEL subscription.

- 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.

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

  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
pem-agent
```

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

```
ppas95-server
ppas-pgpool34
ppas95-pgpool34-extensions
pem-agent
```

For an Advanced Server 9.6 database, the package list must include:

```
edb-as96-server
edb-pgpool35
edb-as96-pgpool35-extensions
pem-agent
```

For an Advanced Server 10 database, the package list must include:

```
edb-as10-server
edb-pgpool36
edb-as10-pgpool36-extensions
pem-agent
```

For a PostgreSQL 9.3 database, the package list must include:

```
postgresql93-server
pgpool-II-93
pem-agent
```

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

```
postgresql94-server
pgpool-II-94
pem-agent
```

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

```
postgresql95-server
pgpool-II-95
pem-agent
```

For a PostgreSQL 9.6 database, the package list must include:

```
postgresql96-server
```

```
pgpool-II-96
pem-agent
```

For a PostgreSQL 10 database, the package list must include:

```
postgresql10-server
pgpool-II-10
pgpool-II-10-extensions
pem-agent
```

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).

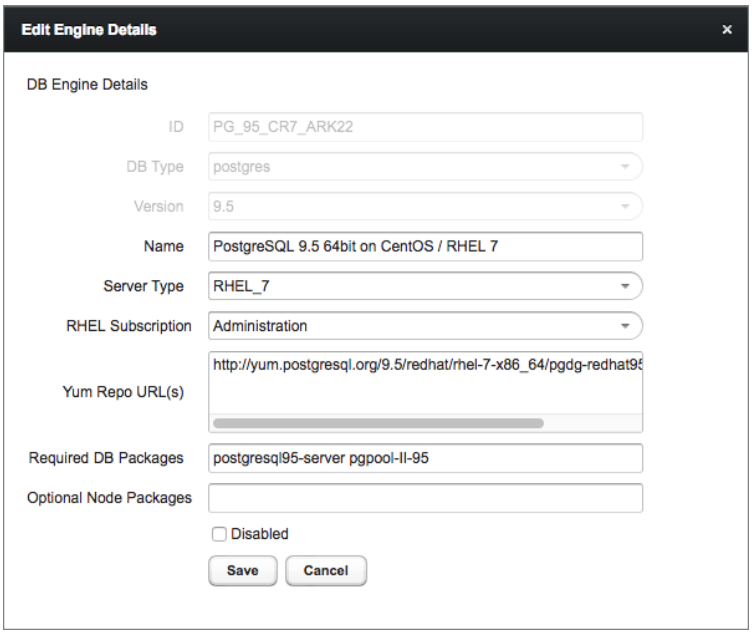

Figure 4.17 – The Edit Engine Details dialog.

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

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.

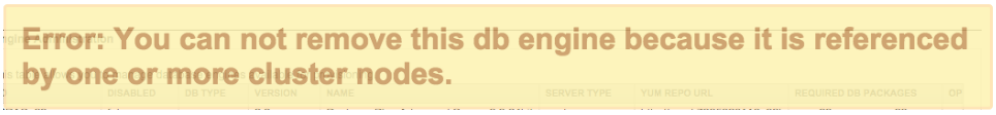

Figure 4.19 – The Delete DB Engine dialog.

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.

4.1.3.2 Adding Supporting Components to a Database Engine Definition

When you create a cluster, you specify the engine that EDB Ark will use when provisioning that 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.

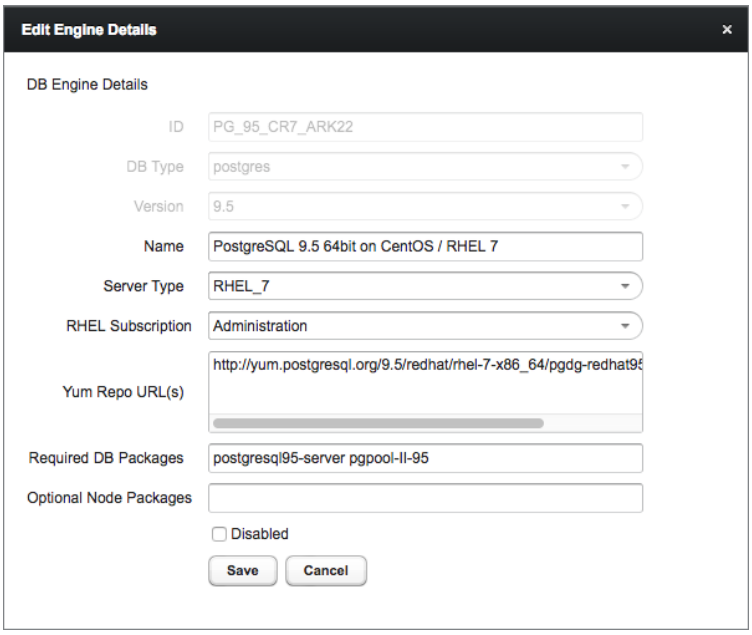

Figure 4.20 – The Edit Engine Details dialog.

4.1.3.2.1 Registering a PEM Agent

The PEM agent is responsible for executing tasks and reporting statistics from a monitored Postgres instance to the PEM server. The PEM agent is installed by the `pem-agent` RPM. By default, all engine configurations shipped with the Ark console include the PEM agent.

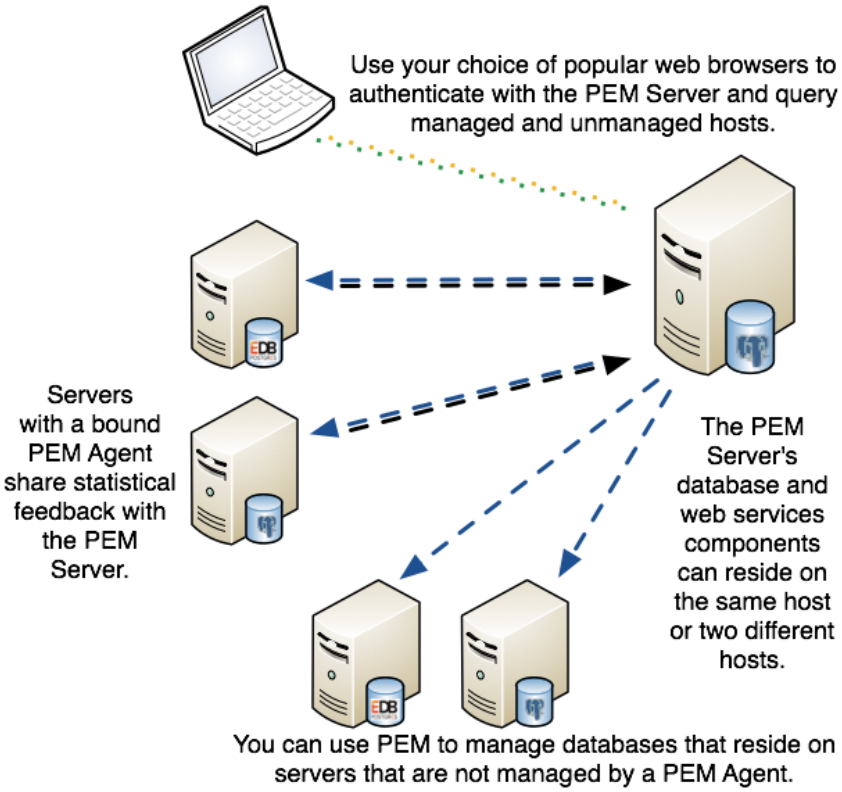

Figure 4.21 – A typical PEM installation.

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 registering the PEM agent with the server, and configuring the server to monitor the agent.

Please note that before registering a node for monitoring, you must:

- modify the `pg_hba.conf` file on the node hosting the PEM 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. The steps assume that you have installed and configured a PEM server; for information about using PEM, please visit the EDB website at:

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

Please note: when a cluster node is stopped (for example, when scaling down), or if a cluster is deleted, the `Monitoring` tab of the PEM web interface will alert you that the agent on that node is down (see Figure 4.22).

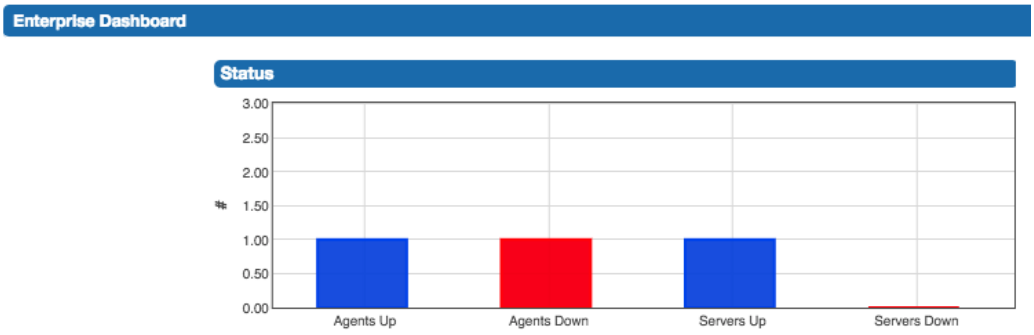

Figure 4.22 – The PEM Enterprise Dashboard displays an agent alert.

If the cluster has been deleted (and the agent will not resume monitoring), you can use the PEM `Browser` tree control to remove the agent definition from the PEM server. Expand the `PEM Agents` node of the tree control, and right-click on the name of the deleted agent; then, select `Delete/Drop` from the context menu.

Step 1 – Create an EDB Ark Cluster

Navigate to the `Clusters` tab, and create a new cluster that is provisioned using an engine definition that includes the `pem-agent` RPM package in the list of required RPM packages.

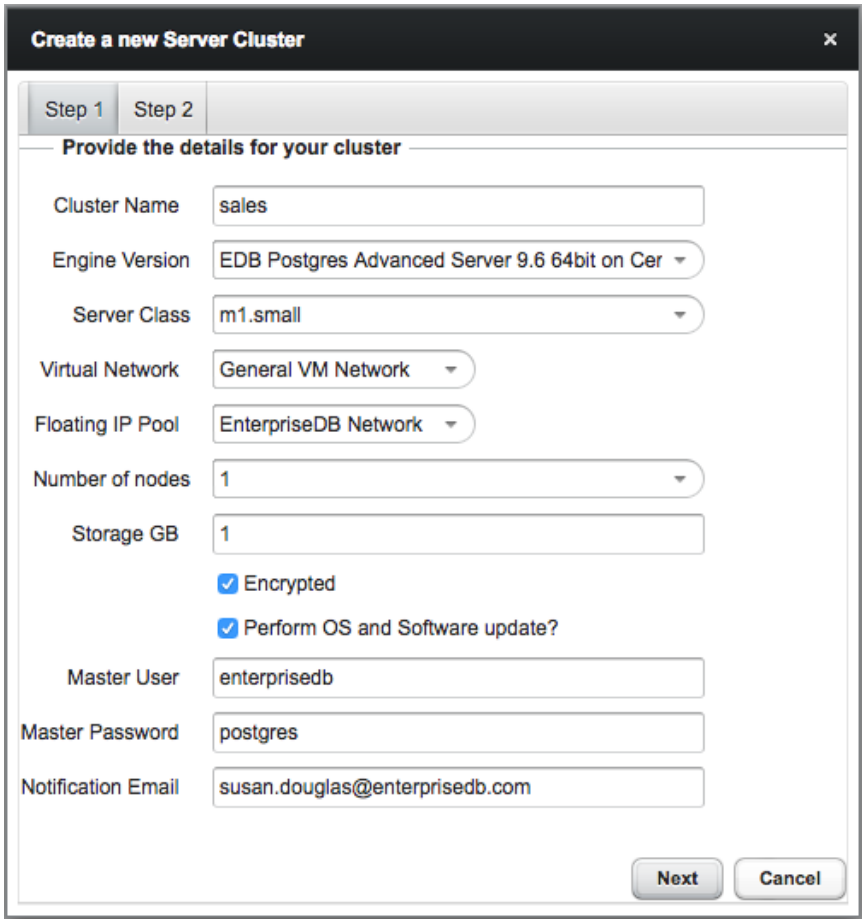

Figure 4.23 – Creating a new Server Cluster

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

Step 2 – 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 web interface. 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 4.24) of the PostgreSQL database that stores PEM statistics to allow connections from any monitored servers as well as the PEM client.

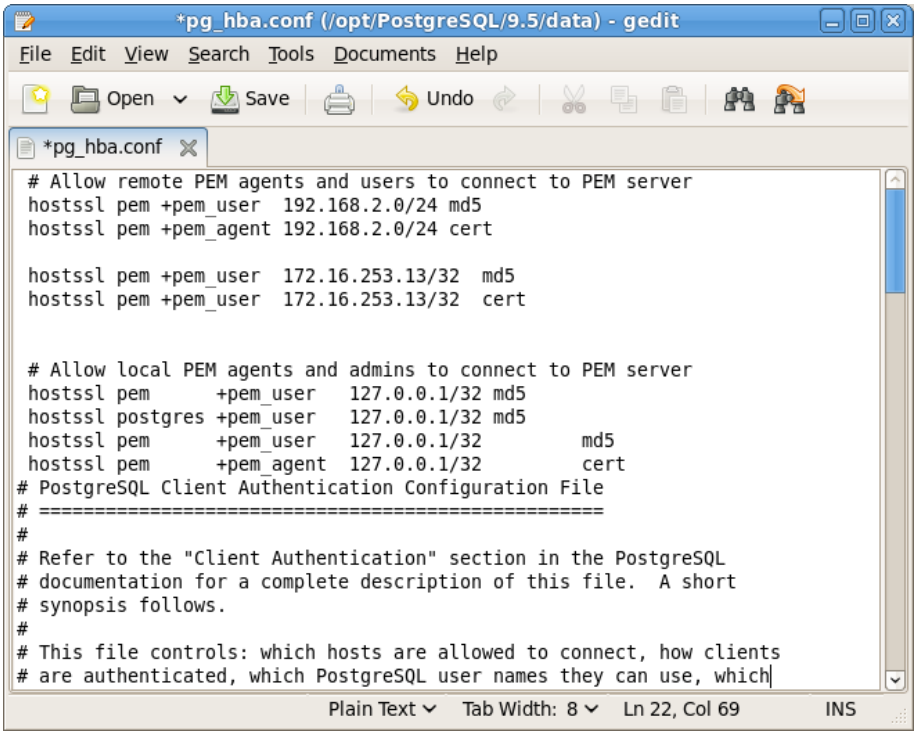

Figure 4.24 – Modifying the PEM Server's pg_hba.conf file.

With your choice of editor, modify the `pg_hba.conf` file of the PEM Server backing database, 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

Step 3 – 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-9.6 restart
```

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

Step 4 – 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 4.25).

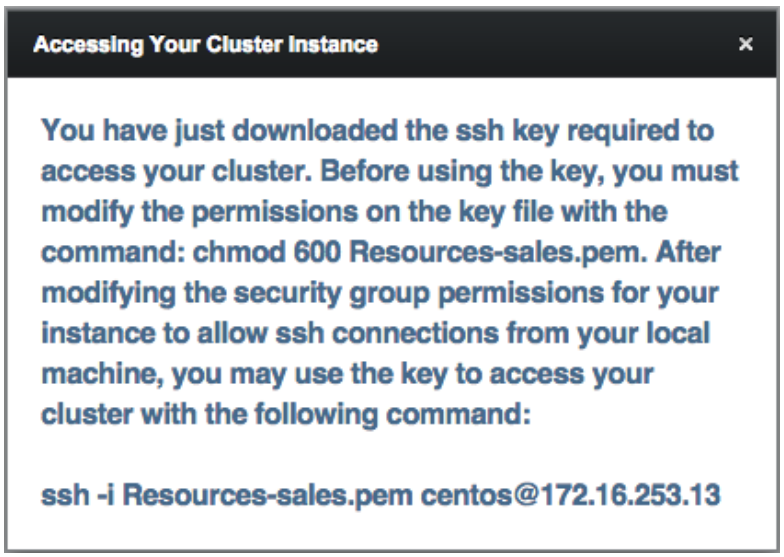

Figure 4.25 - Using SSH to connect to the Ark cluster.

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.

Step 5 – 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 4.26).

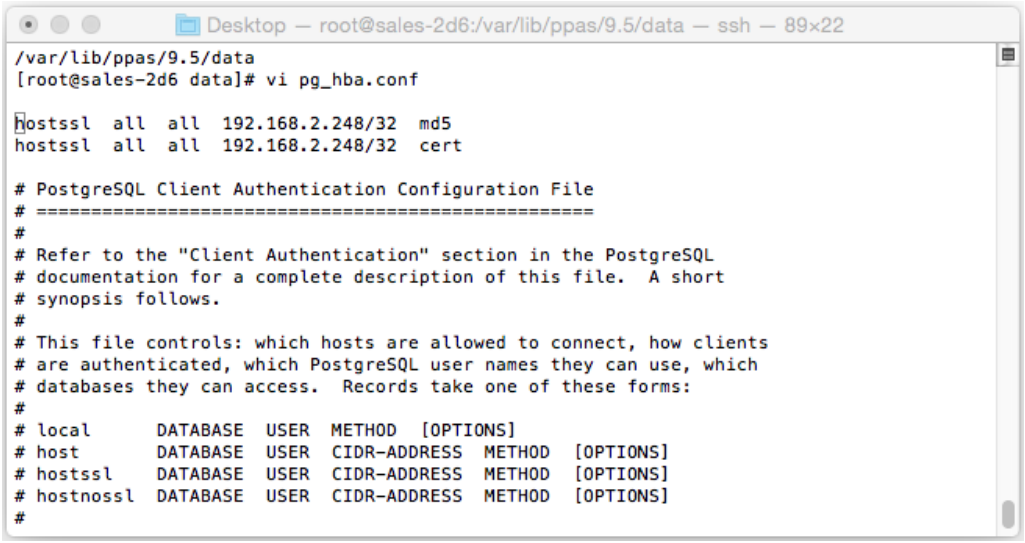

Figure 4.26 – Modifying the Ark cluster's pg_hba.conf file.

Step 6 – 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 7 – 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-7.0/bin
```

Then, invoke the PEM agent registration program:

```
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 4.27).

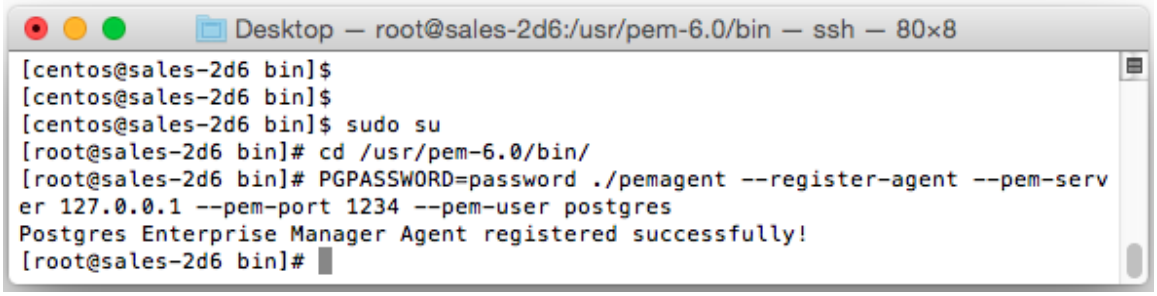

Figure 4.27 – Registering the PEM agent.

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

4.1.3.2.1 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. 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 Administrative 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`.

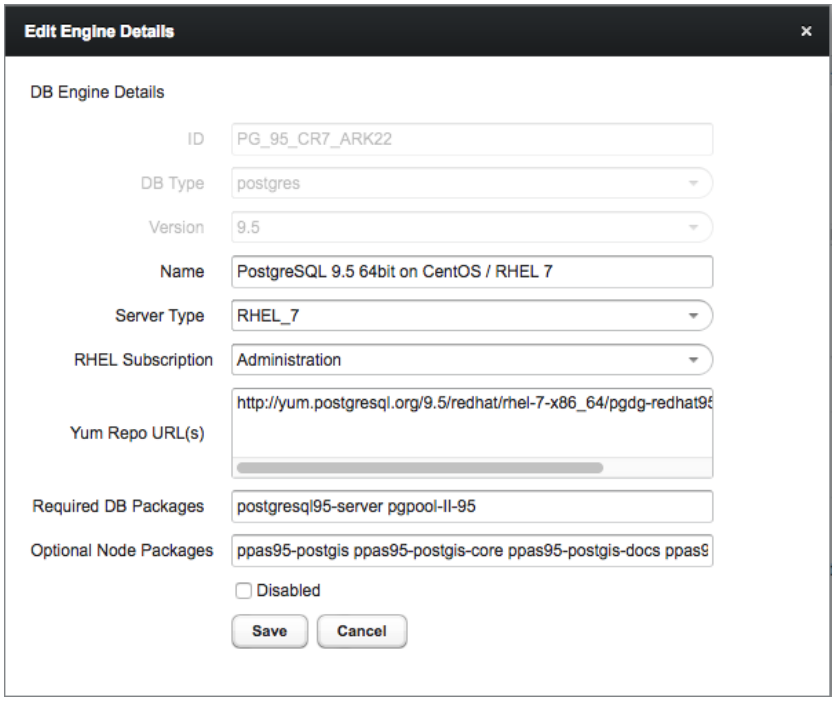

Figure 4.28 – Modifying the Engine Details dialog.

When the `Edit Engine Details` dialog opens (see Figure 4.28), 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
```

The PostGIS installation packages for Advanced Server 9.6 are:

```
edb-as-96-postgis
edb-as-96-postgis-core
edb-as-96-postgis-docs
edb-as-96-postgis-utils
```

The PostGIS installation packages for Advanced Server 10 are:

```
edb-as-10-postgis
edb-as-10-postgis-core
edb-as-10-postgis-docs
edb-as-10-postgis-utils
```

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 4.29).

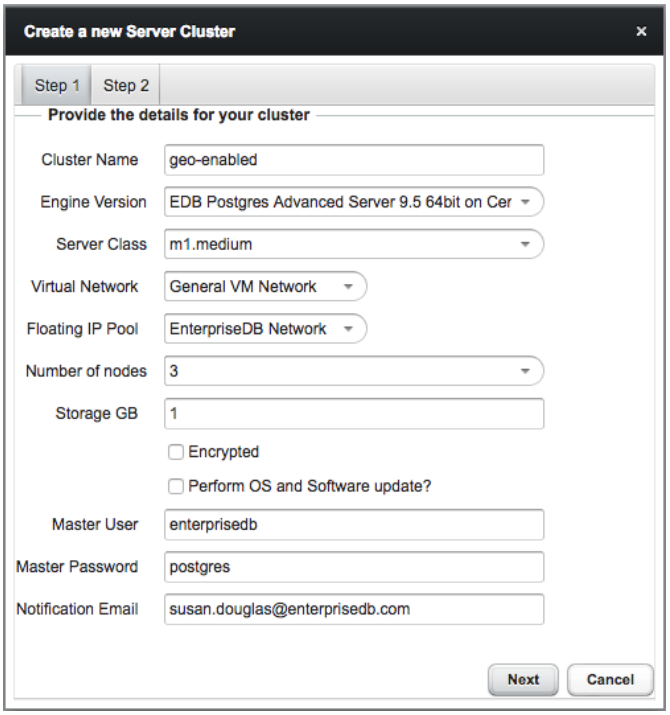

Figure 4.29 – Use the modified engine when provisioning a cluster.

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

Creating the PostGIS Extensions

After adding the packages to the master node of a cluster, you can use the psql client or the EDB Postgres Enterprise Manager (PEM) client to create the extensions. Before connecting with a client, an Administrator must open the listener port (by default, `5444` on an Advanced Server instance) of the node for connections.

Use a client to connect to the database in which you wish to create the extensions, and enter the following commands:

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

The client will confirm that the extensions have been created successfully. The PostGIS functions are created in the `public` schema of the database.

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

http://postgis.net/documentation/

4.1.4 Red Hat Subscription Management

You can use the Ark Administrative console to attach Red Hat Subscription Manager information to engines hosted on Red Hat consoles. The Red Hat Subscription Manager tracks installed products and subscriptions to implement content management with tools like yum. For information about Red Hat Subscription Manager, visit the Red Hat website at:

https://access.redhat.com/documentation/en/red-hat-subscription-management/

When you create a new cluster that uses an engine that is associated with a Red Hat subscription, Ark registers the cluster nodes with Red Hat; when you terminate the node, the system's subcription is unregistered.

Use the `RHEL Subscription Management` section of the `Admin` tab to define and manage Red Hat Subscription Manager access for your Ark consoles that reside on Red Hat Linux instances (see Figure 4.30).

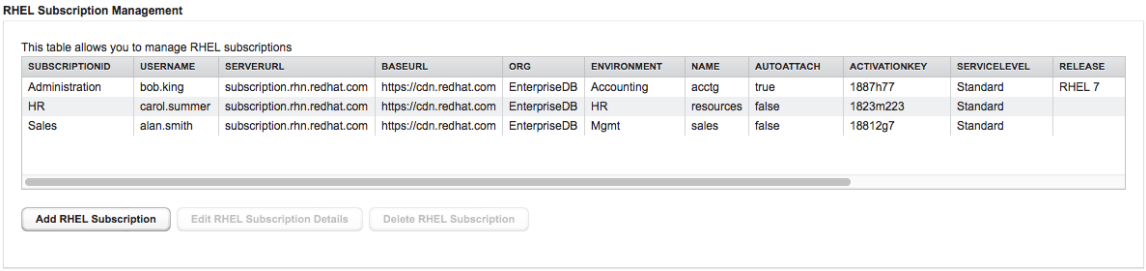

Figure 4.30 – the RHEL Subscription Management section.

After creating a subscription definition, use options in the `DB Engine Administration` section of the `Admin` tab to associate the definition with database engines.

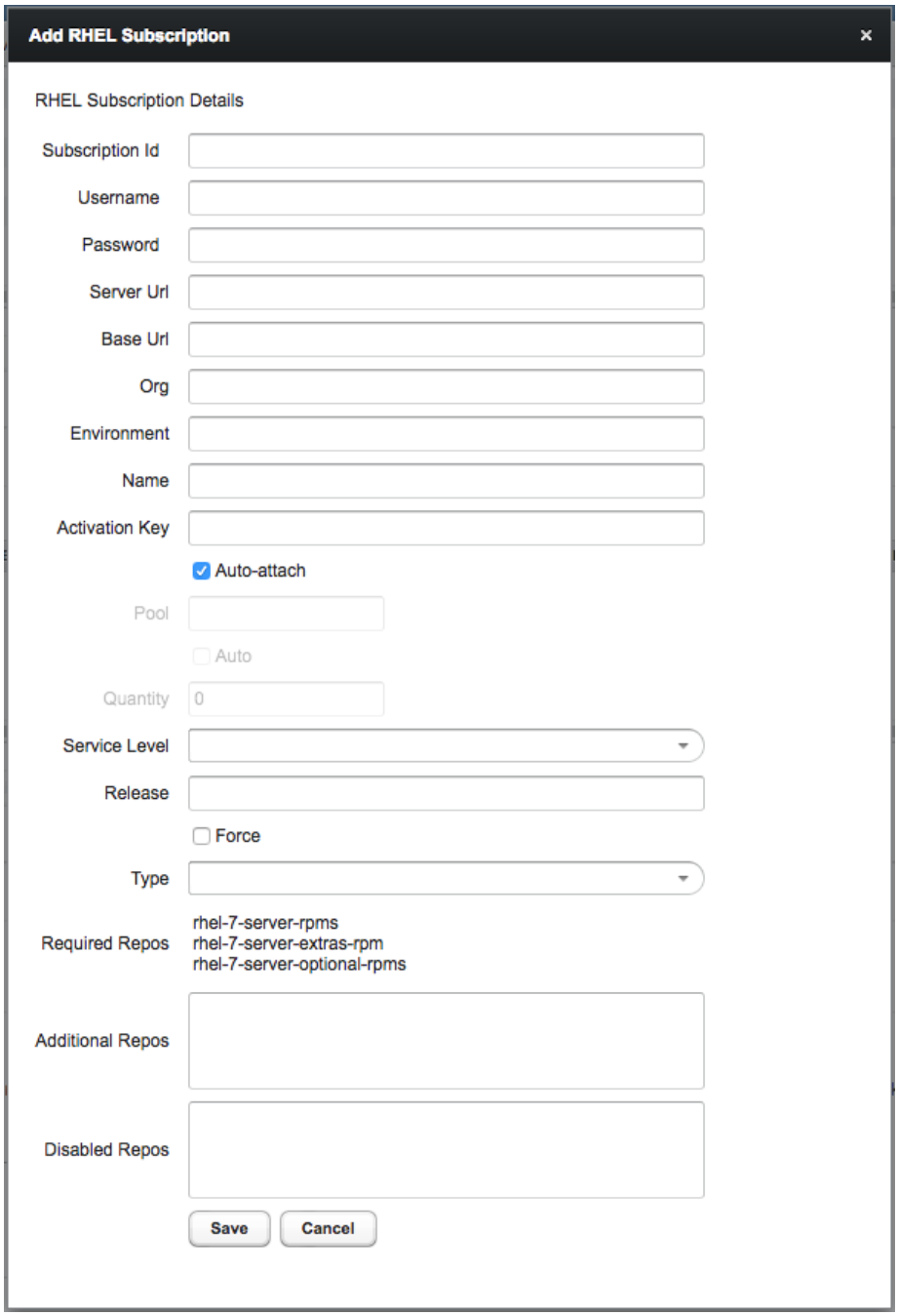

Figure 4.31 – The Add RHEL Subscription dialog.

Use fields on the `Add RHEL Subscription` dialog (see Figure 4.31) to describe a Red Hat subscription service:

- Use the `Subscription Id` field to provide a user-friendly name for the subscription. The name will identify the subscription in the `RHEL Subscription` drop-down on the `Add Engine Details` dialog.

- Use the `Username` field to provide the name of the user account registered with the Red Hat content server.

- Use the `Password` field to provide the password associated with the user account.

- Use the `Server Url` field to provide the host name of the subscription server used by the service; if left blank, the default value of `subscription.rhn.redhat.com` will be used.

- Use the `Base Url` field to provide the host name of the content delivery server used by the service; if left blank, the default value of `https://cdn.redhat.com` will be used.

- Use the `Org` field to provide the organization that will be registered with the Red Hat subscription system.

- Use the `Environment` field to provide the name of the environment (within the organization that will be registered).

- Use the `Name` field to provide the name of the system that will be registered.

- Use the `Activation Key` field to provide the activation key of the Red Hat subscription.

- If enabled, use the `Auto-attach` checkbox to instruct any node associated with the subscription to automatically attach to the service.

- If applicable, use the `Pool` field to provide the pool identifier for the Red Hat subscription service.

- If applicable, check the `Auto` checkbox to indicate that nodes provisioned with engines associated with the pool will automatically attach to the subscription service.

- If applicable, use the `Quantity` field to provide the number of subscriptions in the subscription pool.

- Use the `Service Level` field to provide the service level of the subscription.

- Use the `Release` field to provide the operating system minor release that will be used when identifying updates to any nodes provisioned with the subscription.
- Check the `Force` checkbox to indicate that the node should be registered, even if it is already registered.
- Use the `Type` field to specify the type of consumer that is being registered; the default is `system`.
- The `Required Repos` list is populated by the Ark console, and displays a list of the repositories required by the subscription definition.
- Use the `Additional Repos` field to provide the names of any additional repositories that should be enabled on the cluster node(s).
- Use the `Disabled Repos` field to provide the names of any repositories that should be disabled on the cluster node(s).

When you've completed the dialog, click the `Save` button to add the repository to the table in the `RHEL Subscription Management` section, or `Cancel` to exit without saving. If you choose to save the definition, the Ark console will display a popup that lists the subscription manager commands that were generated as a result of your selections (see Figure 4.32).

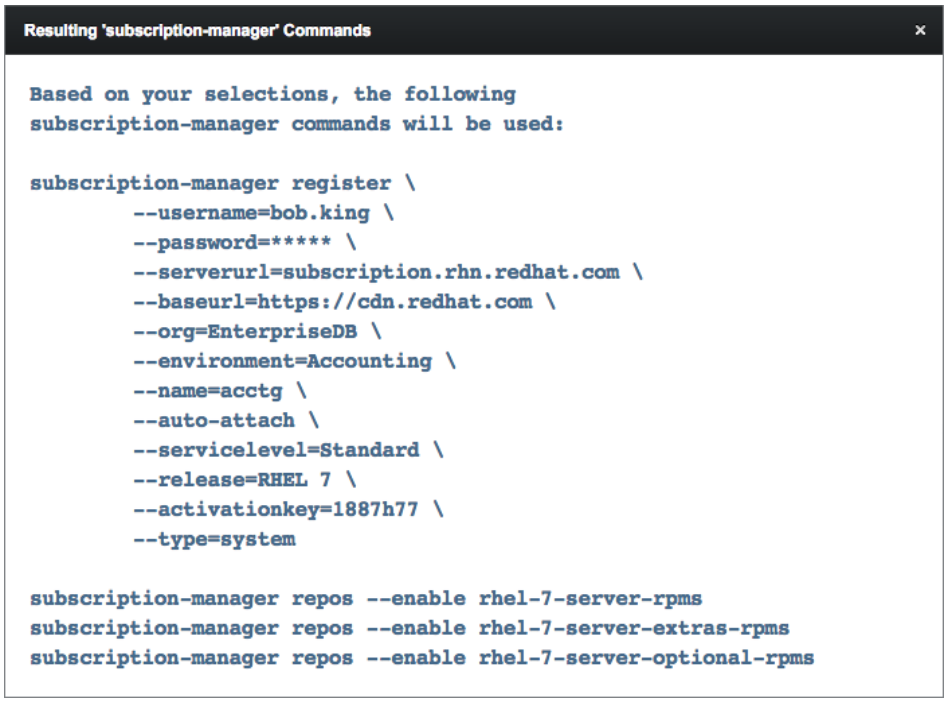

Figure 4.32 – The Add RHEL Subscription dialog.

After creating a subscription definition, use options in the DB Engine Administration section of the Admin tab to associate the definition with database engines; see Section 4.1.3 for detailed information.

Modifying a RHEL Subscription Definition

To modify the description of a Red Hat Subscription Manager service, highlight the name of a subscription in the RHEL Subscription Management table, and click the Edit RHEL Subscription button. The Edit RHEL Subscription Details dialog opens, allowing you to modify the subscription definition.

After modifying the subscription definition, click Save to preserve your changes and exit the dialog; to exit without saving, click the Cancel button. Please note that changes made to a definition are applied only to those instances that are created after the changes are saved; changes are not propagated to existing instances.

Deleting a Red Hat Subscription Definition

Before deleting a Red Hat subscription service definition, you must:

- Modify any database engines that are associated with the subscription, disassociating the engine definition from the Red Hat subscription.
- Delete any instances that were created using an engine that is associated with the Red Hat subscription service.

Then, to delete a Red Hat Subscription Manager service from the list in the Ark console, highlight the name of a service and click the Delete RHEL Subscription button.

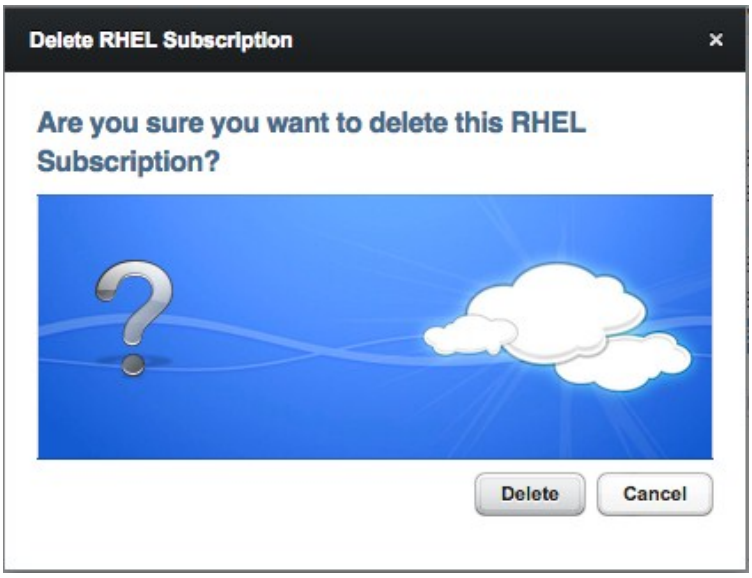

Figure 4.33 – Confirming that you wish to delete a subscription description.

Click the `Delete` button to confirm that you wish to delete the subscription definition, or `Cancel` to exit without deleting the definition (see Figure 4.33).

4.1.5 Managing Amazon Roles

Amazon Role ARNs that are listed in the `IAM Roles Administration` table (see Figure 4.34) will be available on the `Role` drop-down listbox of the `Add User` dialog. Please note that before adding a Role ARN to the table you must define the role in the AWS management console, and the trust policy of the role must include the `External Id` of the Ark console.

Figure 4.34 – The Roles Administration dialog.

You can use the `Add Role` dialog to add an entry to the table. To locate the information required by the `Add Role` dialog, connect to the Amazon Management dashboard, and navigate to the `Roles` page. Select the role you wish to add from the list to open the `Summary` dialog; then, select the `Trust relationships` tab to display the information required (circled in red in Figure 4.35).

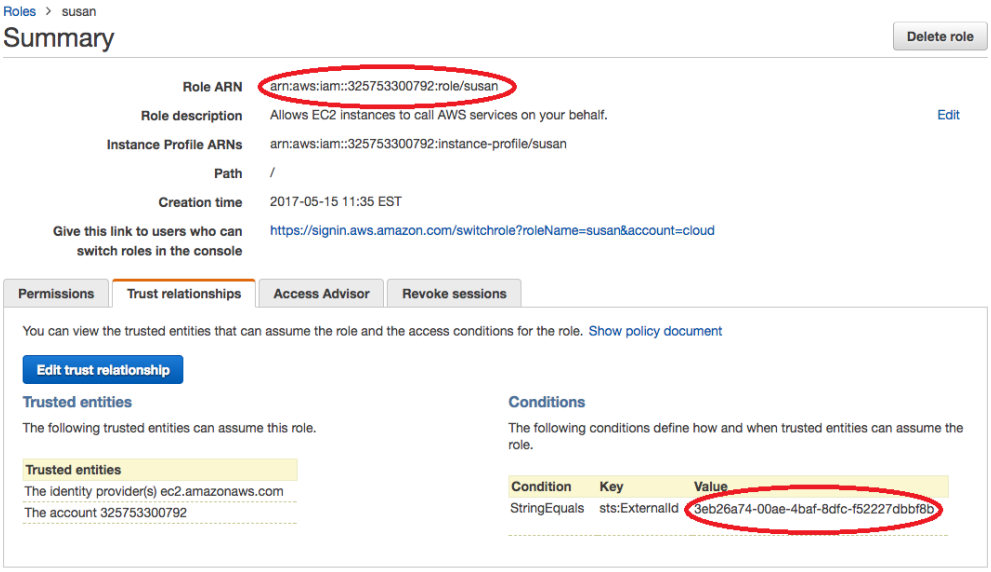

Figure 4.35 – The Roles Administration dialog.

To add a Role ARN to the table, click the Add Role button; the Add Role dialog opens as shown in Figure 4.36.

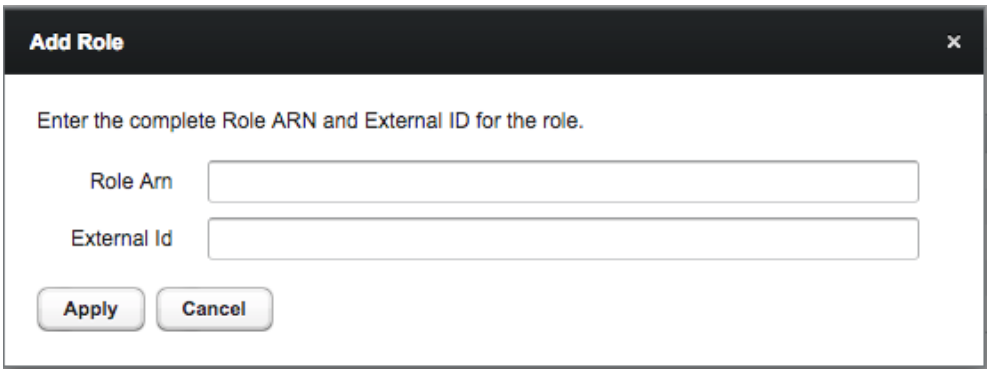

Figure 4.36 – The Roles Administration dialog.

Use fields on the Add Role dialog to provide details from the Amazon management console:

- Provide the Role ARN from the Summary dialog header in the Role Arn field.
- Provide the Value from the Trust relationships tab in the External Id field.

Click the Apply button to verify the information, and add the entry to the table.

4.1.6 User Administration

Options in the `User Administration` section of the `Admin` tab provide extended management functionality for an administrative user. The functionality offered is host and configuration specific.

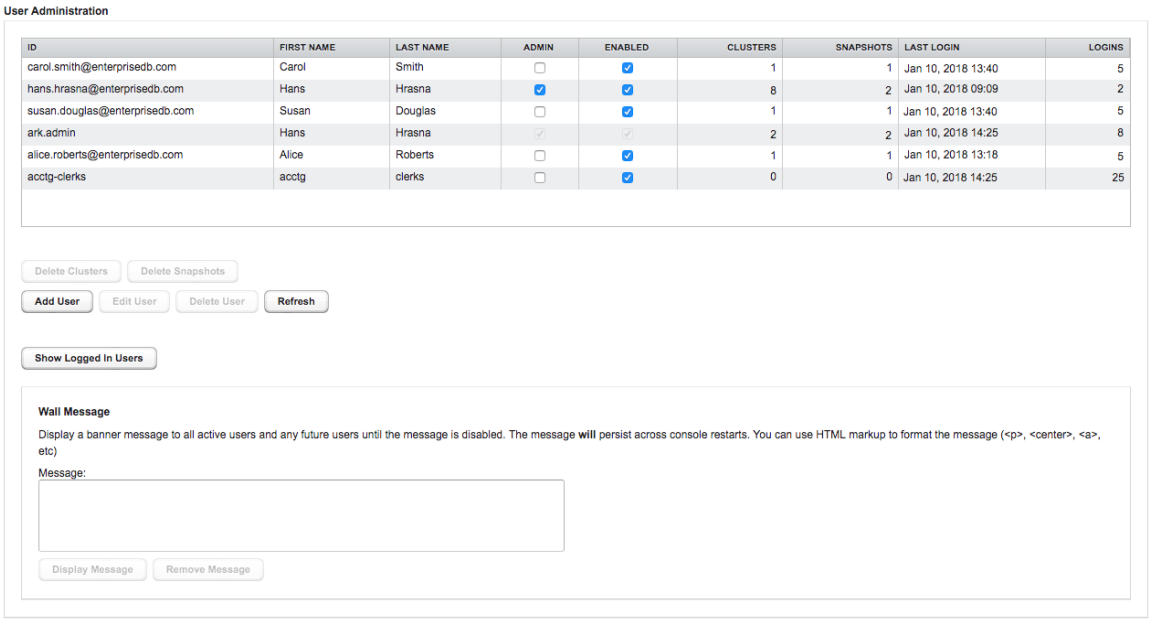

Figure 4.37 –User administration features of the Amazon console.

Depending on your host type and configuration, you can use `User Administration` options to:

- add, modify, or delete a user account.
- delete clusters or snapshots that belong to a user account.
- display a list of logged in users.
- add, modify, or remove a wall message.

Adding a User

If available for your configuration, you can click the `Add User` button to access the `Add User` dialog (see Figure 4.38) and register a new user account for the Ark console.

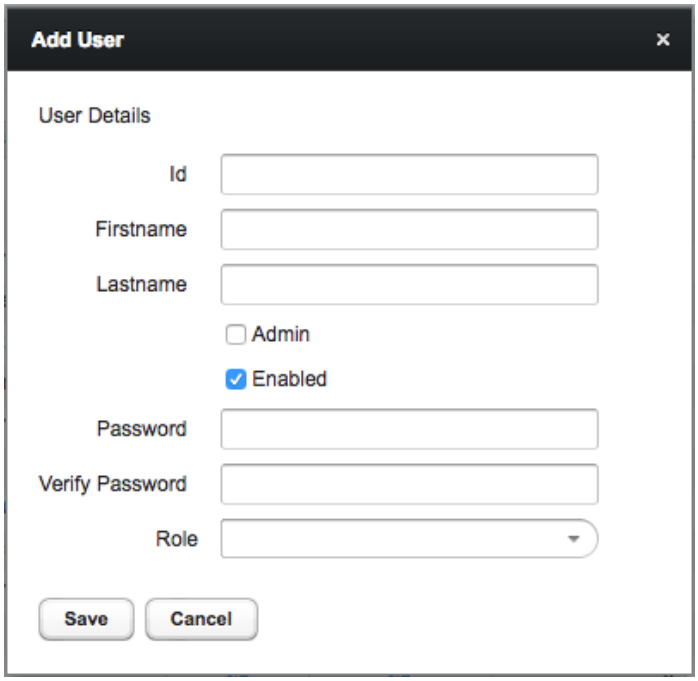

Figure 4.38 – The Add User dialog.

Provide information about the 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`.
- Check the box next to `Enabled` if the user should be allowed to log in to the console.
- If applicable, provide a password associated with the user account in the `Password` field.
- If applicable, confirm the password in the `Verify Password` field.
- If applicable, 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 User Properties and Reviewing User Activity

If the `Edit User` button is displayed, you can use the `Edit User` dialog to modify user properties. Highlight a user name, and click the `Edit User` button to open the `Edit User` dialog (see Figure 4.39).

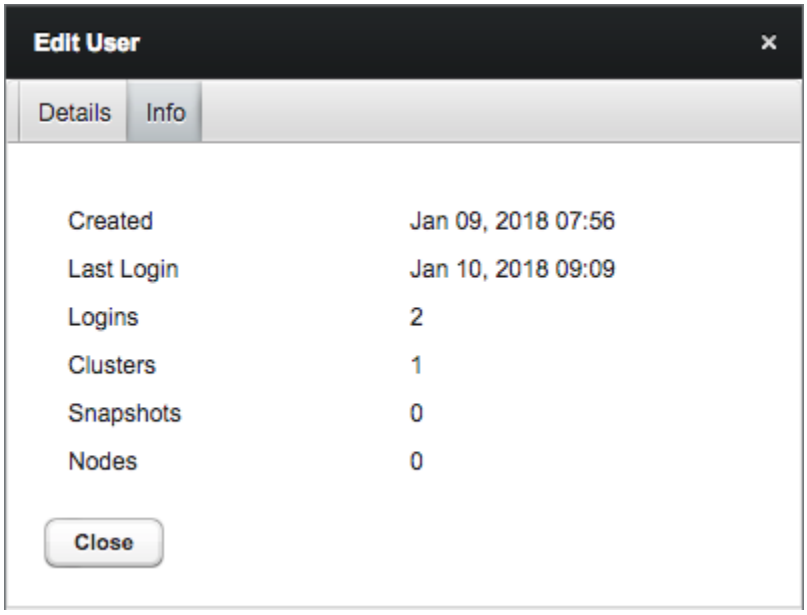

Figure 4.39 – The Edit User dialog.

Enabled fields on the `Details` tab may be modified; use the `Info` tab to review information about the user account and account activities.

Deleting User Objects

If displayed, you can use buttons below the `User Administration` table to manage user objects. Highlight a user name, 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.

After deleting the objects owned by a user, the `Delete User` button will remove the user account. To delete a user, highlight the name of a user in the user table, and click the `Delete User` button. The Ark console will ask you to confirm that you wish to delete the selected user before removing the account. Click `Delete` to remove the user account, or `Cancel` to exit the popup without deleting the account.

4.1.3.3 User Administration on an Amazon Host

When deployed to use Postgres authentication on an Amazon host, the `User Administration` tab will display the `User Administration` table. You can use the `User Administration` table to register new users for the Ark console, edit user properties, or delete a user account (see Figure 4.40). Please note that you still must use a client application to connect to the Ark console and add the user to the `postgres` database before the user is allowed to connect.

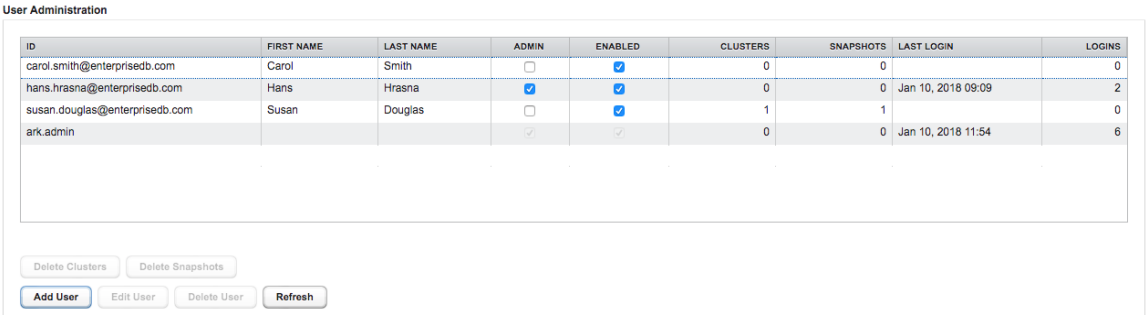

Figure 4.40 – The user table of an AWS console

Columns within the `User Administration` 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.

- 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.

4.1.3.4 User Administration on an Azure Host

When deployed to use Postgres authentication on an Azure host, the `User Administration` tab will display the `User Administration` table. You can use the `User Administration` table to register new users for the Ark console, edit user properties, or delete a user account (see Figure 4.41). Please note that you still must use a client application to connect to the Ark console and add the user to the `postgres` database before the user is allowed to connect.

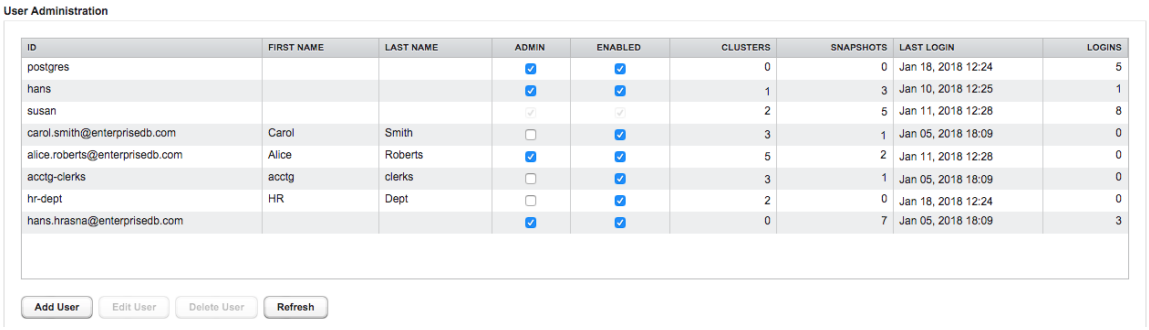

Figure 4.41 –User administration features of the Azure console.

Use the check boxes to modify user access privileges:

- Check the box next to a user name in the `ADMIN` column to indicate that the user should have administrative access to the Ark console.

- Check the box next to a user name in the `ENABLED` column to indicate that the account is active.

Use the `Refresh` button to update the `User Administration` table.

4.1.3.5 User Administration on an OpenStack Host

When deployed to use Postgres authentication on an OpenStack host, the `User Administration` tab will display the `User Administration` table. You can use the `User Administration` table to register new users for the Ark console, edit user properties, or delete a user account (see Figure 4.42). Please note that you still must use a client application to connect to the Ark console and add the user to the `postgres` database before the user is allowed to connect.

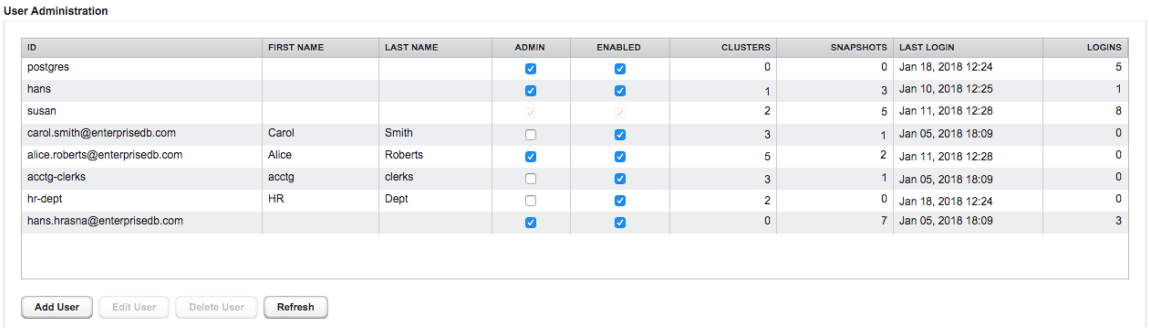

Figure 4.42 –User administration features of the Azure console.

Use the check boxes to modify user access privileges:

- Check the box next to a user name in the `ADMIN` column to indicate that the user should have administrative access to the Ark console.

- Check the box next to a user name in the `ENABLED` column to indicate that the account is active.

Use the `Refresh` button to update the `User Administration` table.

4.1.3.6 Displaying Connected Users

Click the `Show logged in users` button to display the `Logged in users` dialog (see Figure 4.43).

Figure 4.43 – 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.

4.1.3.7 Managing the Wall Message

Provide a message in the `Message` field (shown in Figure 4.44) 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.

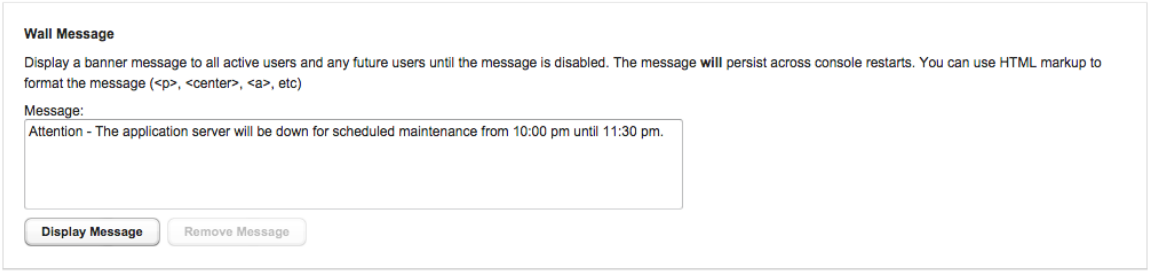

Figure 4.44 - Modifying the Wall Message.

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 4.45).

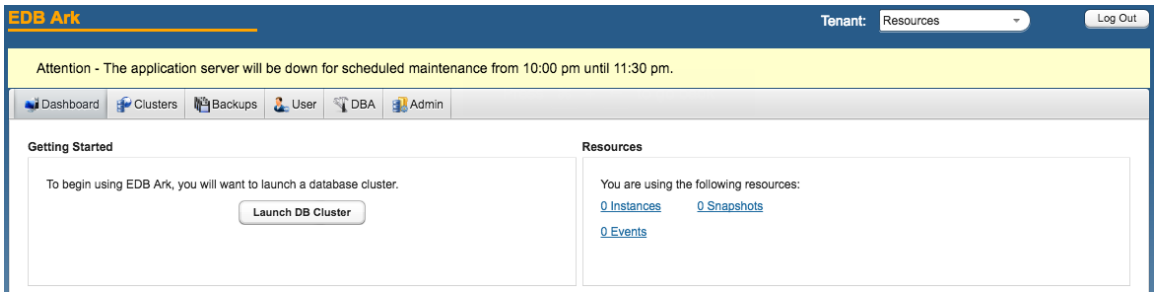

Figure 4.45 - Displaying a wall message.

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.

4.1.7 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 4.46).

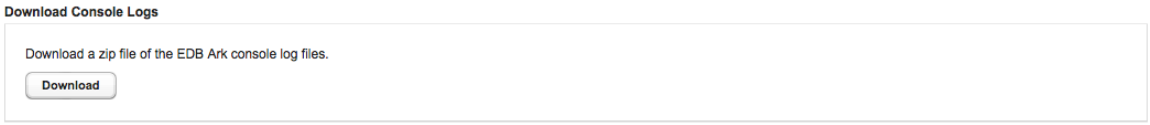

Figure 4.46 – The user table of an AWS console.

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:

```
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 4.47) 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.

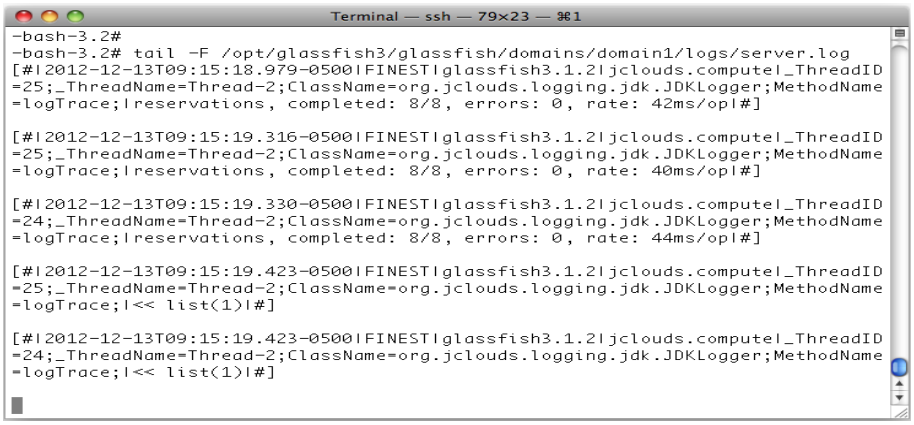

Figure 4.47 - Following the log file with the tail utility.

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

```
tail -F file_name
```

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

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

```
tail –help
```

4.1.8 Editing Installation Properties

Use the option displayed in the `Edit Installation Properties` section to review or modify Ark console properties (see Figure 4.48).

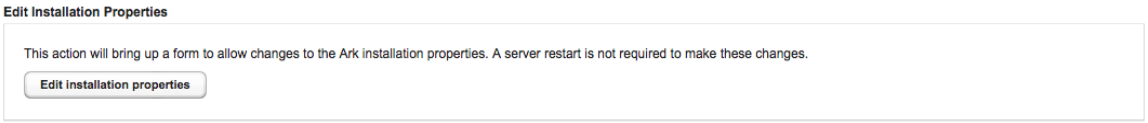

Figure 4.48 – The Edit Installation Properties section.

Click the `Edit installation properties` button to open the `Edit Installation Properties` dialog (see Figure 4.49). Use fields on the `Edit Installation Properties` dialog to modify the properties of the Ark console. For detailed descriptions of each field:

- For an Amazon-hosted console, see Section 3.1.3.
- For an OpenStack-hosted console, see Section 3.2.6.
- For an Azure-hosted console, see Section 3.3.5.

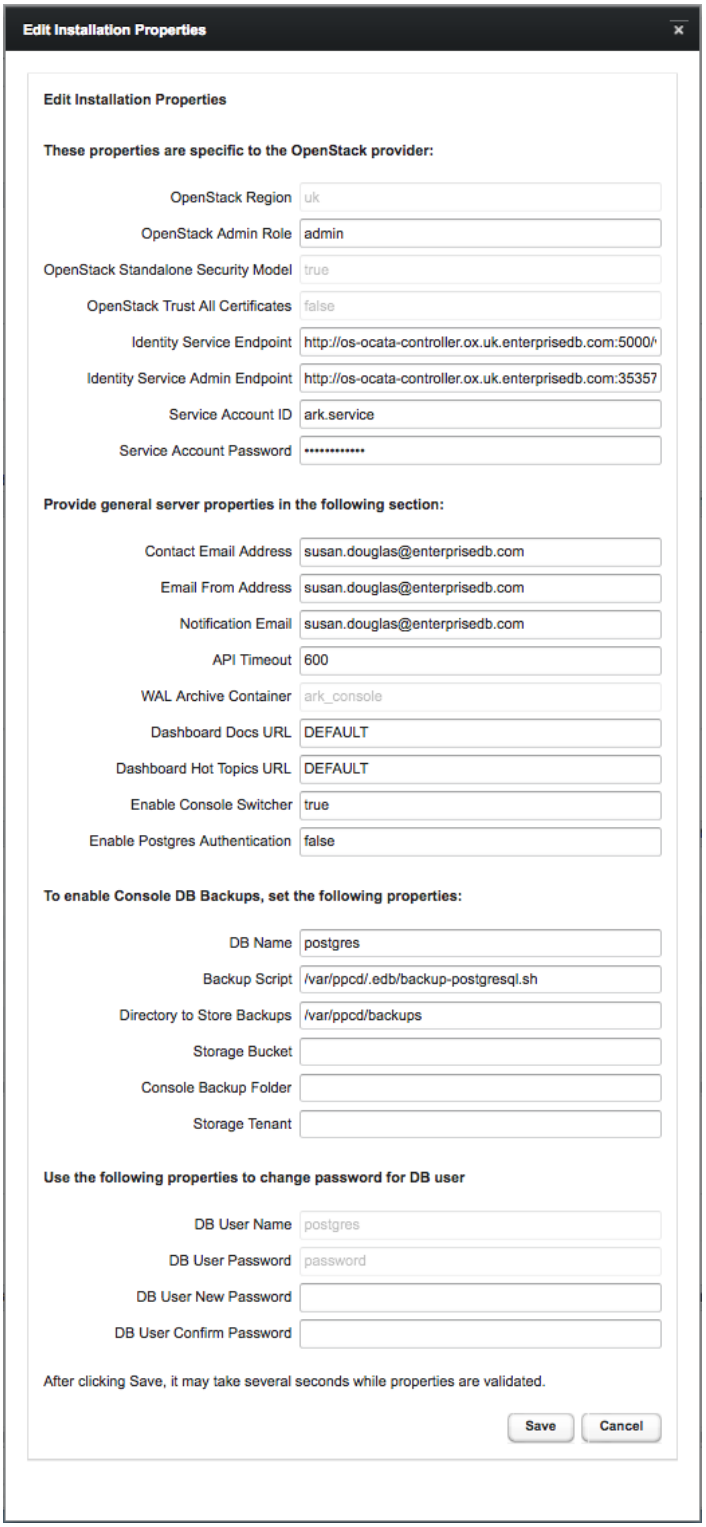

Figure 4.49 – The Edit Installation Properties dialog.

When you've finished, click `Save` to preserve your changes and restart the console server, or `Cancel` to exit the dialog without saving the changes.

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 4.50) is accessible only to administrative users.

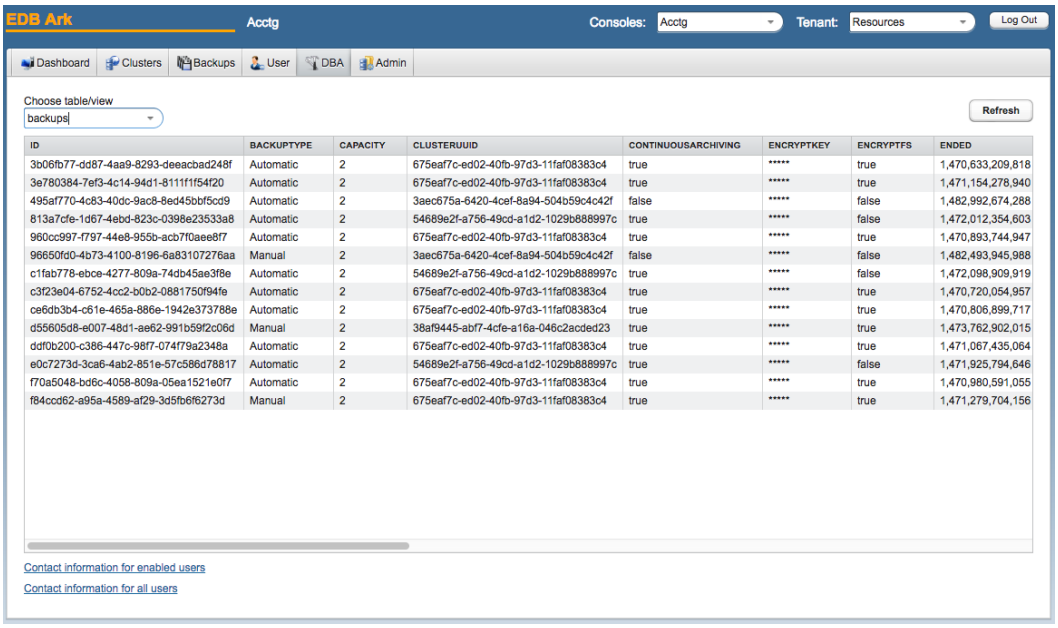

Figure 4.50 - The DBA tab.

Use the `Choose table/view` drop down listbox (shown in Figure 4.51) to select a view.

Figure 4.51 - The table/view listbox.

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

Accessing User Information

Use the user information links in the lower-left corner of the `DBA` tab (shown in Figure 4.52) to download a comma-delimited list of users and user information.

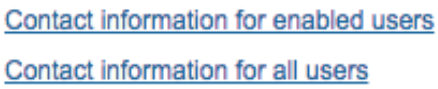

Figure 4.52 - The contact information links

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

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

Select a link to download user information:

- Click `Contact information for enabled users` to download a file that contains only those users that are currently enabled.

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

4.3 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.

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.

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.
RHELSUBSCRIPTION_ID	The identifier of the Red Hat subscription associated with the 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.

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.

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.

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 name of a property.
VALUE	The value of the property.

4.3.10 rhelrepo

The `rhelrepo` table provides information about the repositories required by the described Red Hat Subscription Manager services.

Column Name	Description
ID	The unique identifier of the repository.
REPO_NAME	The repository name.
SUBSCRIPTION_ID	The descriptive identifier of the Red Hat Subscription Manager service.

4.3.11 rhelsubscription

The `rhelsubscription` table provides information about currently defined Red Hat Subscription Manager services.

Column Name	Description
ID	The unique identifier of the server.
ACTIVATION_KEY	The activation key of the Red Hat subscription.
AUTO_ATTACH	Indicates if nodes associated with the subscription will automatically attach to the service.
BASE_URL	The content delivery server used by the service.
ENVIRONMENT	The name of the environment (within the organization that will be registered).
FORCE	Indicates if the node should be registered (even if the node is already registered).
NAME	The name of the system that will be registered.
ORG	The organization that will be registered with the Red Hat subscription system.
PASSWORD	The password associated with the user account.
RELEASE	The operating system minor release that will be used when identifying updates to any nodes provisioned with the subscription.
SERVER_URL	The host name of the subscription server used by the service.
SERVICE_LEVEL	The service level of the Red Hat subscription.
SUBSCRIPTION_ID	The user-friendly name of the subscription.
TYPE	The type of consumer that is being registered by the subscription service.
USERNAME	The name of the user account registered with the Red Hat content server.
POOL	The pool identifier for the Red Hat subscription service.
QUANTITY	The number of subscriptions in the subscription pool.
ATTACH_AUTO	Indicates if nodes using the pool will automatically attach to the service.

4.3.12 serverimage

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

Column Name	Description
`ID`	The unique identifier of the server.
`IMAGE_ID`	The OpenStack identifier of the server image.
`INIT_USER`	The virtual machine OS user (as provided on the `Add Server` dialog).
`SERVER_DESCRIPTION`	The server description.
`SERVER_ID`	The descriptive identifier of the server.
`OS_TYPE`	The operating system type of the server.
`IS_STATIC`	The provisioning mode of the server; true if the server is static, false if the server is not static.

4.3.13 snapshots

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

Column Name	Description
`ID`	The unique snapshot identifier.
`BACKUPID`	An application-managed foreign key reference to the ID column of the `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.

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 `SSL` encryption.

5.1 Modifying a Security Group for an OpenStack Hosted Console

Before a user may SSH to 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, open the `Project` menu, expand the `Network` menu, and select `Security Groups`. 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).

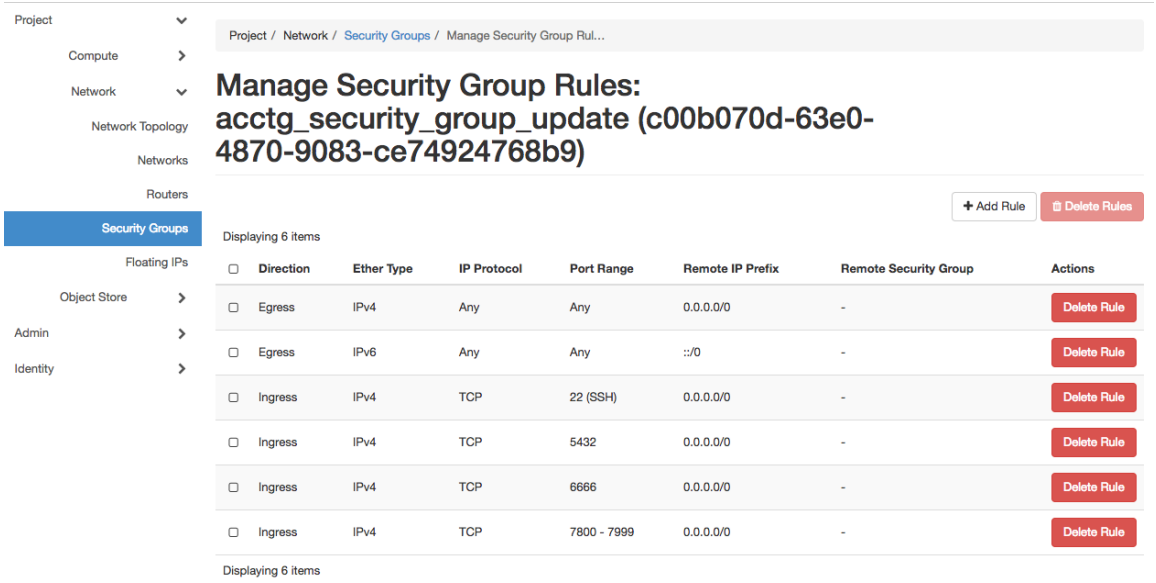

Figure 5.1 – Detailed security rules for a cluster.

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`.

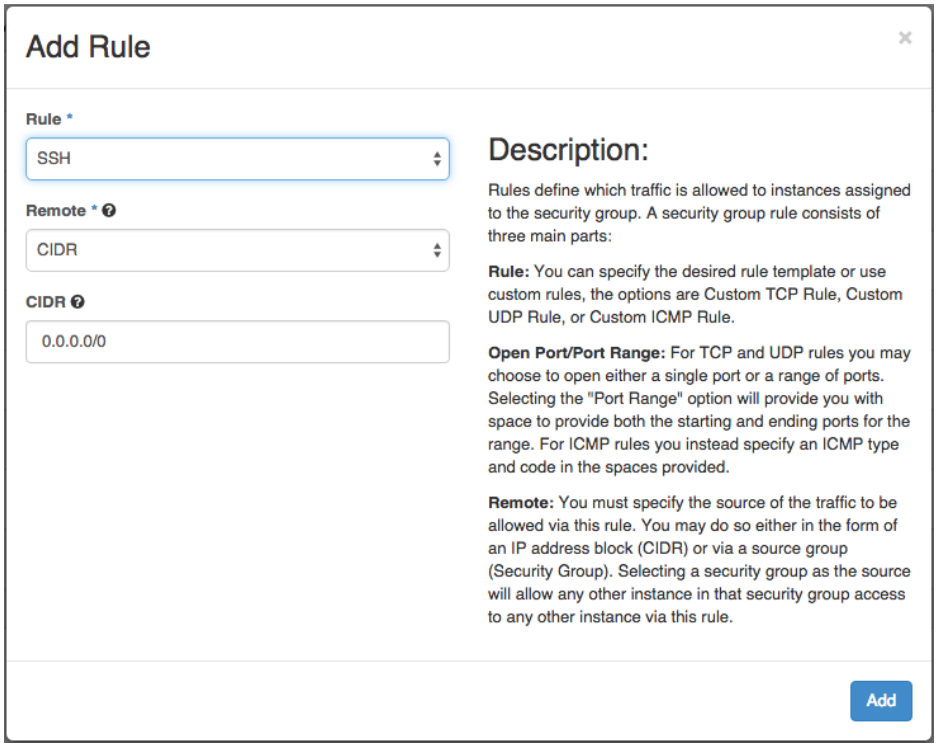

Figure 5.2 – Opening a port for an SSH connection.

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.

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).

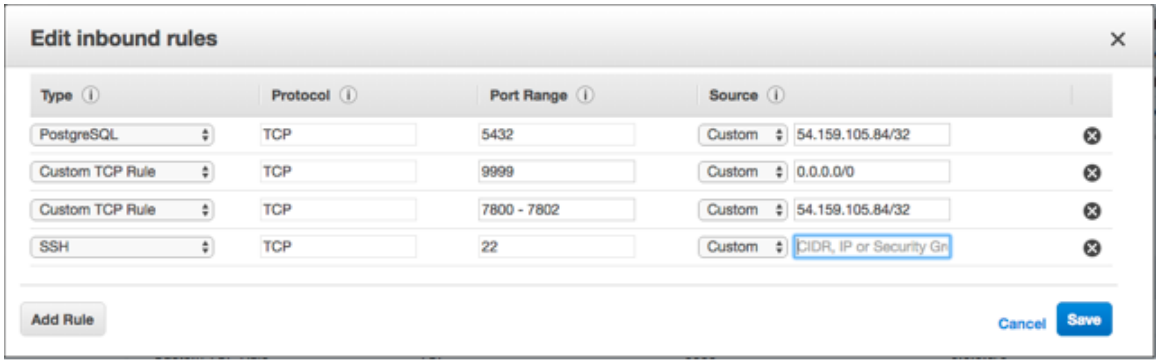

Figure 5.3 – Opening a port for an SSH connection.

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

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 5.4).

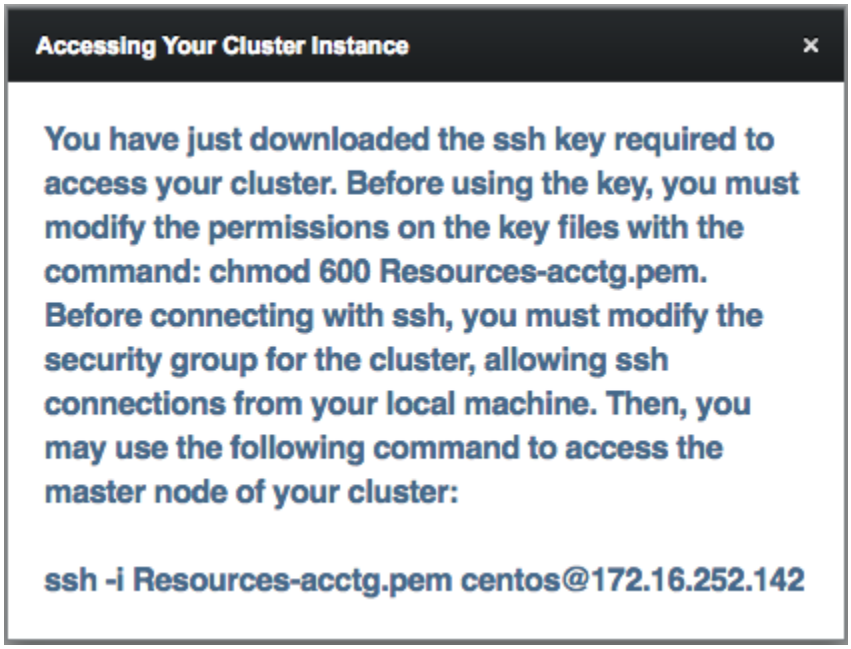

Figure 5.4 – Accessing Your Cluster Instance.

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.

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
```

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

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` on CentOS hosts, or `cloud-user` on RHEL hosts) 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 installer-creaeted user should lose ssh access for any reason.

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 6.3.

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 4.1.5.

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. Then:

1. Use `ssh` to connect to the node on which the Ark console resides, and assume root privileges:

   ```
   sudo su -
   ```

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

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

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

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

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

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

   ```
   yum update package_name
   ```

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

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

   ```
   cd /var/ppcd
   ```

6. Invoke the EDB Ark post-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.

If the Ark software locates an existing `ppcd.properties` file, the configuration values are written into the Ark console database, and the old file is renamed to `ppcd.properties_old_`*`timestamp`*.

During the upgrade process, the Ark RPM is careful not to overwrite any existing files that have changed. The package manager 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 parameter values) 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.

6.3 Changing Console Passwords

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 password on the PostgreSQL server to a unique password (known only to trusted users). You can set the password when you deploy the console or modify the password later on the `Edit Installation Properties` dialog. To open the `Edit Installation Properties` dialog, navigate to the `Admin` tab of the Ark console and click the `Edit Installation Properties` button.

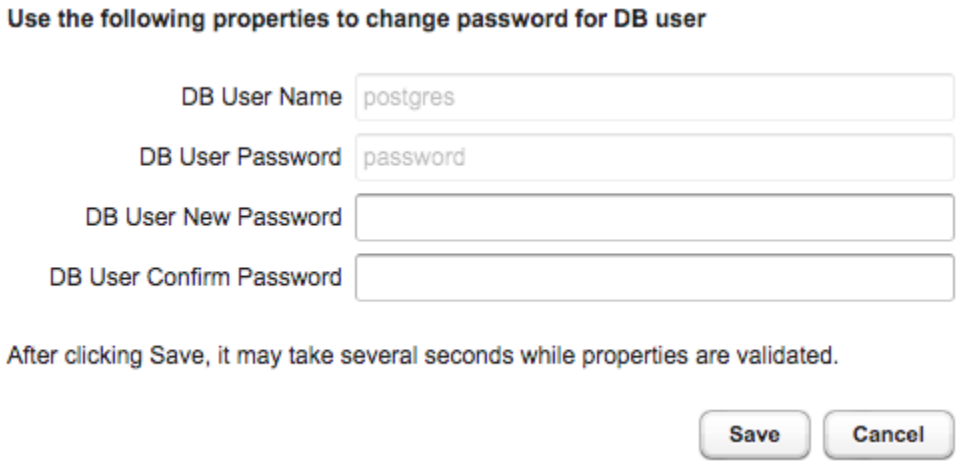

Figure 6.1 – Modifying the database password.

Fields near the bottom of the dialog allow you to modify the password (see Figure 6.1):

- Use the `DB User New Password` field to modify the database password.
- Use the `DB User Confirm Password` field to confirm the new password.

After providing a new password and confirming the password, click the `Save` button. The console will inform you that it needs to restart the server to complete the password change.

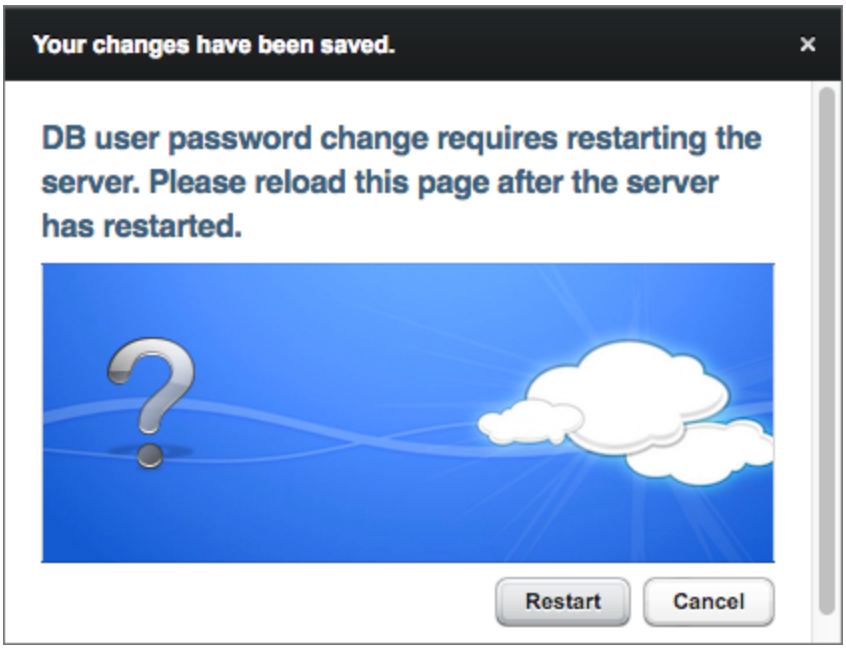

Figure 6.2 – Modifying the database password.

Click the `Restart` button. When the restart is complete, you will be required to log in to the server again.

Modifying the GlassFish Console Password

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

name: `admin`
password: `ChangeIt2015!`

To modify the password associated with the GlassFish user, use ssh to connect to the console image, authenticating yourself with the account id and key pair used when the instance was created. 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.
```

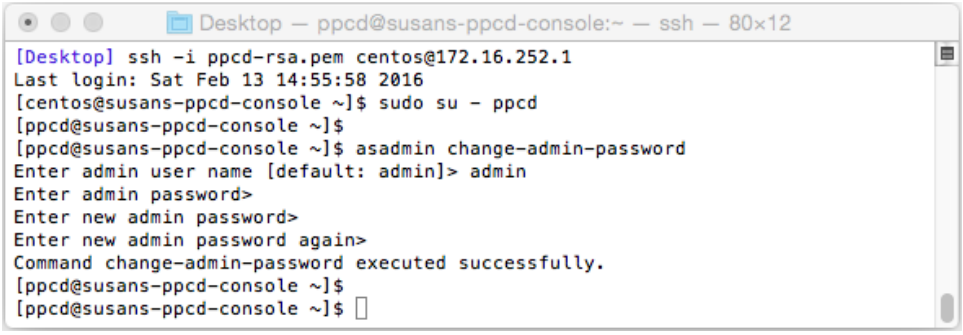

Figure 6.3 – Changing the console user's password.

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:

```
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 (see Figure 6.2):

```
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.
```

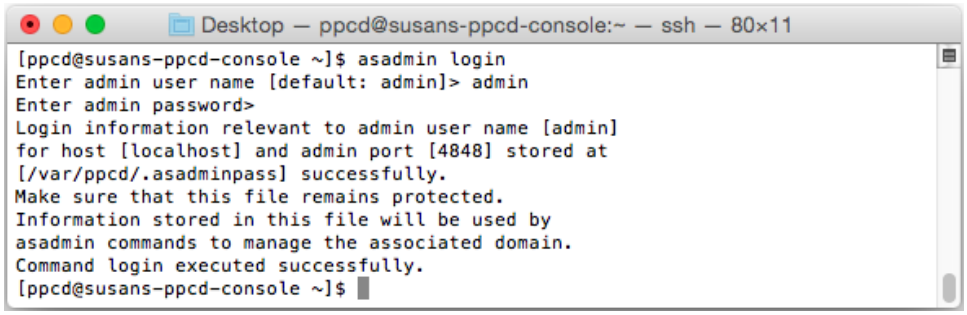

Figure 6.4 – Invoking asadmin login.

6.4 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/WEB-INF/classes/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
```

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.

6.5 Importing SSL Certificates on OpenStack

If your Ark console resides on an OpenStack host that enables HTTPS endpoints, you must import the OpenStack SSL certificates to the Ark's Glassfish web server. Please note that you must import the certificates immediately after the Ark instance is started, and before configuring the console. To import a certificate:

1. Connect to the OpenStack host as the Ark Administrative user (`centos`). Then, write a list of accepted certificates to the `certs.txt` file:

   ```
   sudo openssl s_client -showcerts -connect OpenStack_URL:443
   > certs.txt
   ```

   Where *`OpenStack_URL`* specifies the URL of the OpenStack host.

2. Convert the file to DER format, and write the output to `certs.der`:

```
sudo openssl x509 -in certs.txt -out certs.der -outform DER
```

3. Use the `keytool` command to read the content of the `certs.der` file into a keystore named `clopenstack`:

   ```
   sudo keytool -v -alias clopenstack -importcert -file
   certs.der -keystore
   /opt/glassfish3/glassfish/domains/domain1/config/cacerts.jk
   s
   ```

   When prompted, provide a password; the default password is `changeit`.

4. Use the following commands to restart the Glassfish server:

   ```
   sudo su ppcd

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

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.

The Ark console is configured to take automatic backups of the console database hourly, and after the registration of each new user. If you do not wish to use the Ark backup script to implement backups, you should maintain regular backups of your console database.

7.1 Modifying Backup Properties with the EDB Ark Console

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. Privileges for the `ppcd` user are managed in the `.pgpass` file. The `.pgpass` file (used for backup authentication) is located in the `ppcd` user's home directory (`/var/ppcd`).

You can use the parameters on the `Installation Properties` dialog to modify console backup details; to modify the properties on an installed console, navigate to the `Admin` tab, and click the `Edit Installation Properties` button.

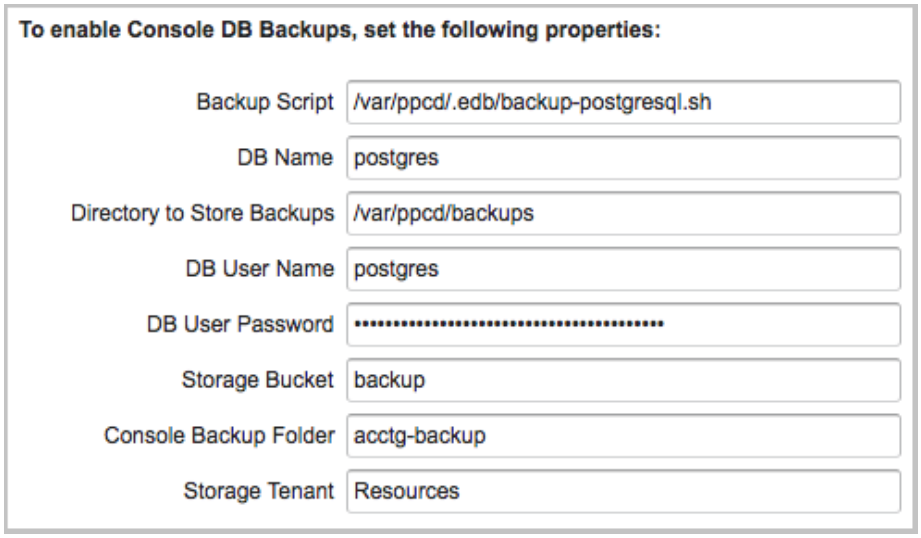

Figure 7.1 – The console backup properties.

When the Edit Installation Properties dialog opens, you can modify details about the console backup storage (see Figure 7.1):

- Use the Backup Script field to specify the name and location of the backup script provided with EDB Ark. If you choose to provide your own backup script, use the parameter to specify the name and location. Please note that you must ensure that the script can be read and executed by the ppcd user.

- Use the DB Name field to specify the name of the console database; the default is postgres.

- Use the Directory to Store Backups field 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.

- Use the DB User Name field to specify the name of the console database user; the default is postgres.

- Use the DB User Password field to specify the password associated with the console database user; the default password is:

  0f42d1934a1a19f3d25d6288f2a3272c6143fc5d

- Use the Storage Bucket field to specify the name of the swift storage container that will be used to store files for point-in-time recovery. This location should not change after the initial deployment of the Ark console.

- Use the Console Backup Folder field to specify a folder in which the backups will be stored.

- Use the Storage Tenant field to provide the name of the tenant in which the backup will be stored.

7.1.1 Using the Recover Option on an AWS Backed Console

If the console cannot locate a registered user, and your console is configured to support console backups, the Ark console login dialog will request the password specified during setup and display the `Deploy Console` or `Recover from Backup` options when you navigate to the console address (see Figure 7.2).

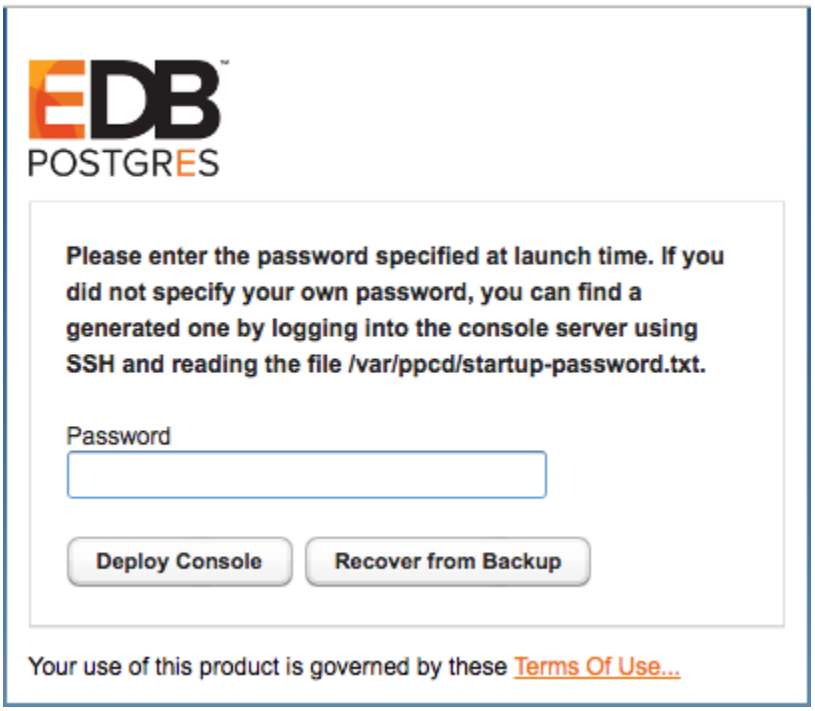

Figure 7.2 - The connection dialog.

To initiate a console recovery, provide the console password specified when you deployed the console instance (in the Amazon management console), and click the `Recover from Backup` button. The console properties dialog opens, prompting you for information about console backups (see Figure 7.3).

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.

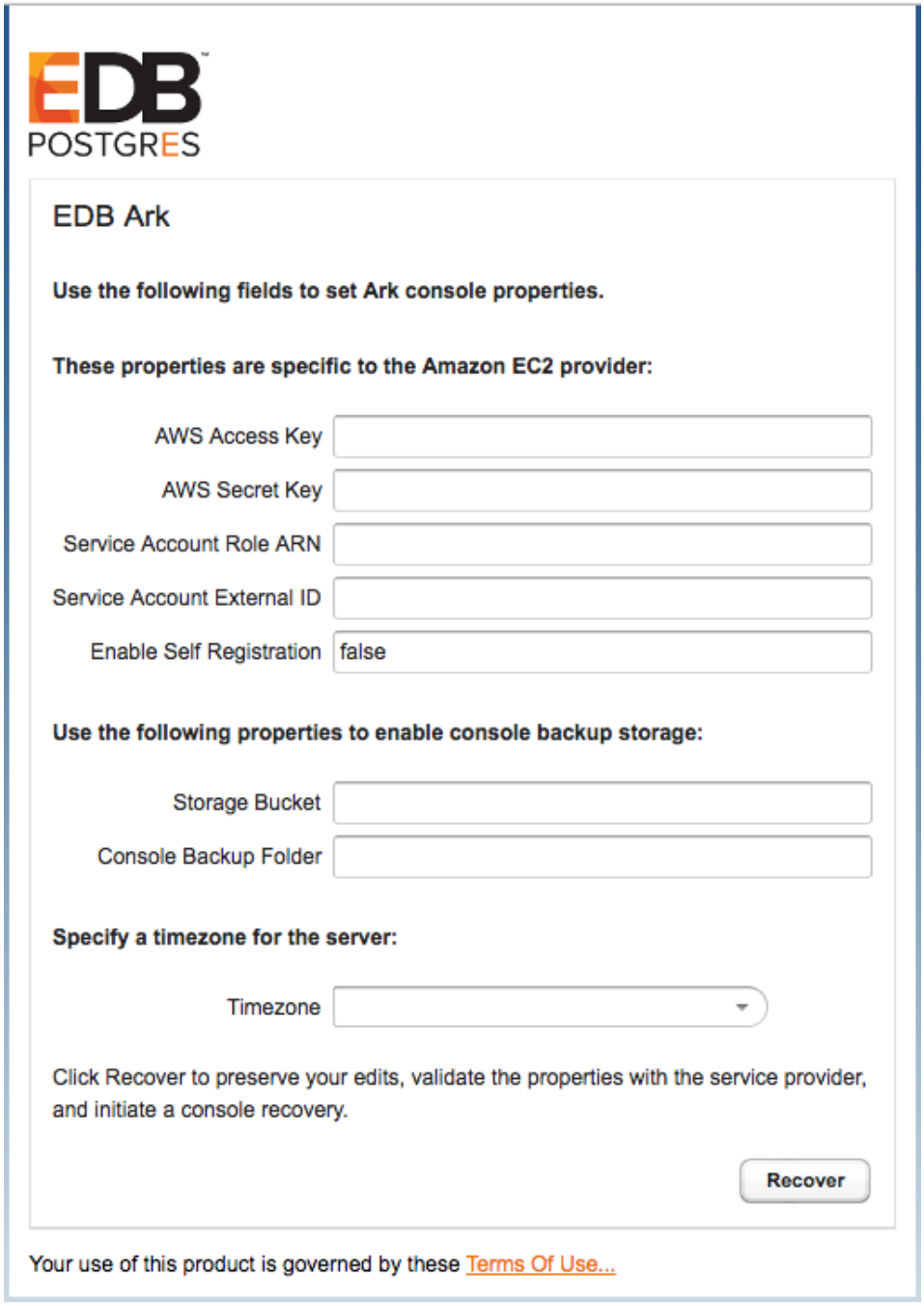

Figure 7.3 – Provide the console properties.

Provide details about the console, and the location of a backup to recover:

- Use the `AWS Access Key` field to specify the Amazon access key ID associated with the AWS role that will be used for account administration.

- Use the `AWS Secret Key` field to specify the Amazon secret key associated with the AWS role that will be used for account administration.

- Use the `Service Account Role ARN` field to specify the Amazon Role ARN (resource name) that should be used by the Ark service user (`ppcd`) when performing management functions on behalf of Ark.

- Use the `Service Account External ID` field to specify the Amazon external ID that should be used by the Ark service user (`ppcd`).

- Use the `Enable Self Registration` field to specify if the Ark console should allow self-registration for Ark users; specify `true` to allow self-registration, or `false` to disable self-registration.

Use the next section to specify your console backup storage details:

- Use the `Storage Bucket` field to specify the name of the bucket in which backups will be stored.

- Use the `Console Backup Folder` field to specify the name of the backup folder within the storage bucket.

Use the last field to specify a timezone for the server:

Use the drop-down listbox in the `Timezone` field to select the timezone that will be displayed by the Ark console.

When you're finished, click the `Recover` button to start the recovery process. A popup will open, prompting you for the name of the backup folder that you wish to use for the recovery.

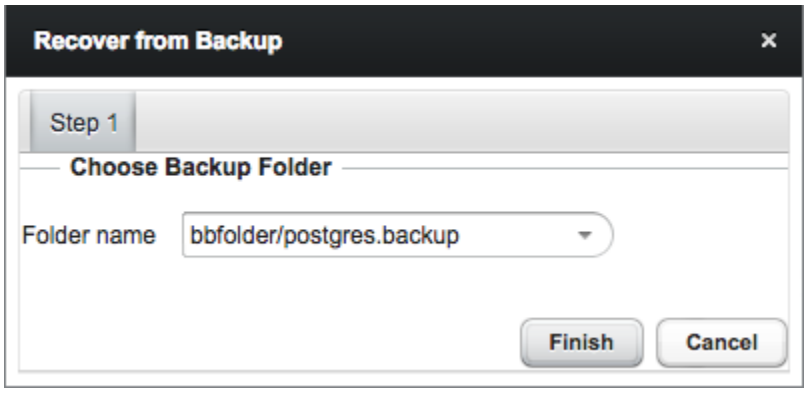

Figure 7.4 – Select a backup folder.

Use the `Folder name` drop-down listbox (see Figure 7.4) to select the backup you wish to use for the recovery, and click `Finish` to start the recovery process.

7.1.2 Using the Recover Option on an OpenStack Backed Console

If the console cannot locate a registered user, and your console is configured to support console backups, the Ark console login dialog will request the password specified during setup and display the `Deploy Console` or `Recover from Backup` options when you navigate to the console address (see Figure 7.5).

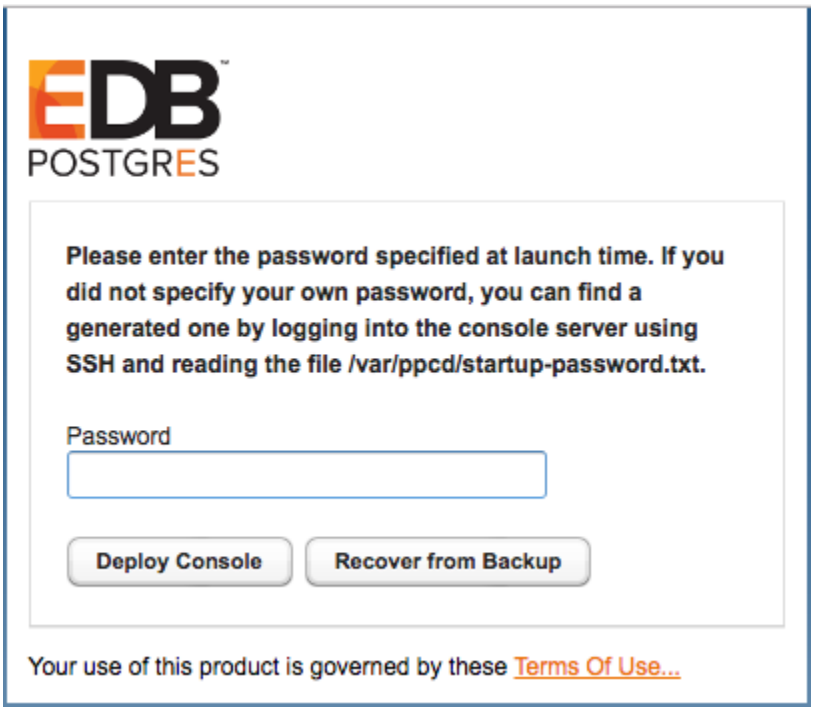

Figure 7.5 – Provide a password to Recover from Backup.

To initiate a console recovery, provide the console password specified when you deployed the console instance (in the OpenStack Management console), and click the `Recover from Backup` button. The console properties dialog opens, prompting you for information about console backups (see Figure 7.6).

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.

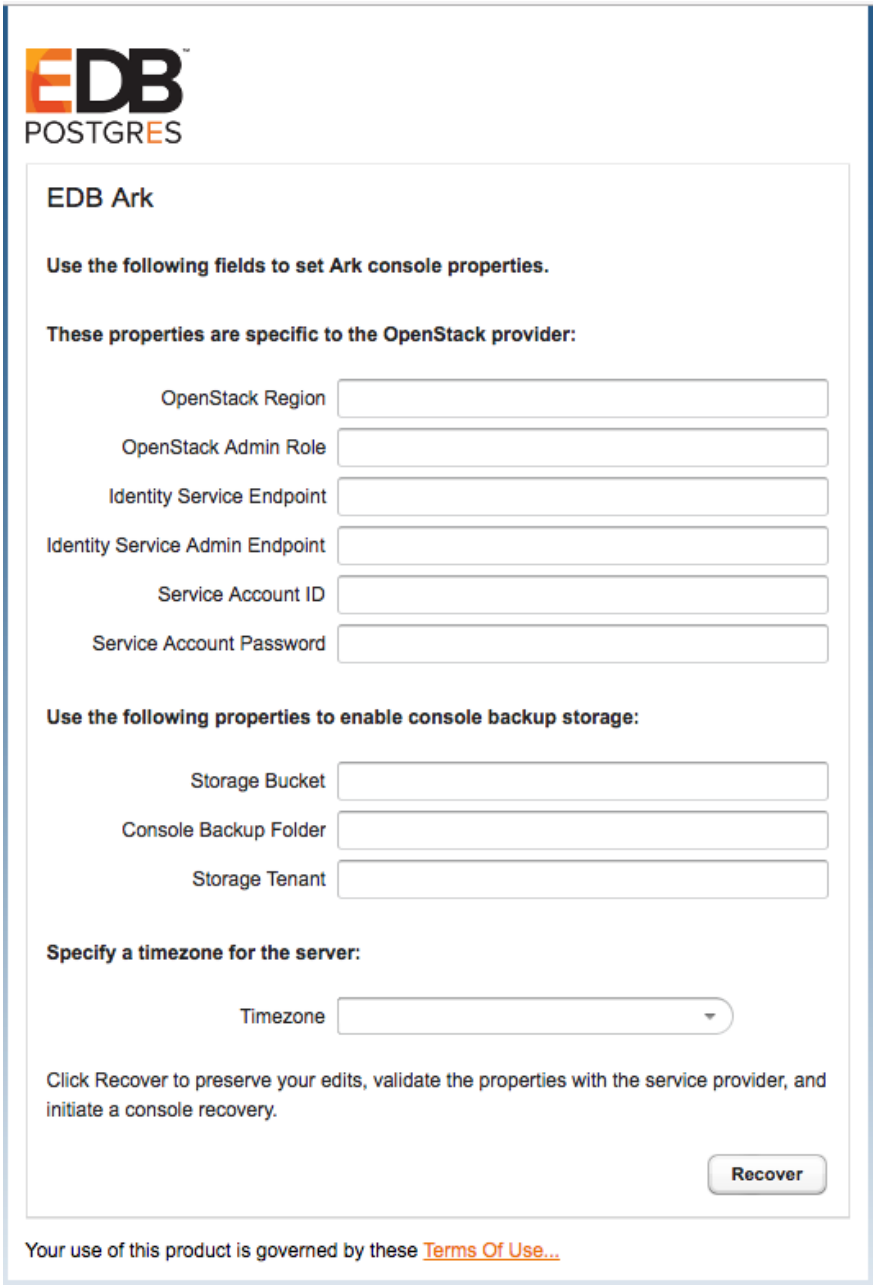

Figure 7.6 – Provide information about a console backup to recover.

Provide details about the console, and the location of a backup to recover:

- Use the `OpenStack Region` field to specify the region in which the OpenStack host resides.

- Use the `OpenStack Admin Role` field to specify the name of the OpenStack administrative role. When a user that is a member of this role connects to the

console, the console will display the Ark administrative console (which includes the Admin and DBA tabs).

- Use the Identity Service Endpoint field to specify the URL of the OpenStack Keystone Identity Service.

- Use the Identity Service Admin Endpoint field to specify the URL of the OpenStack Keystone Administrative Service.

- Use the Service Account ID field to specify the name of the OpenStack user account that Ark will use when managing clusters. The account must be a member of and be assigned the admin role for all tenants that are allowed to run Ark clusters.

- Use the Service Account Password field to specify the password associated with the OpenStack service account.

Provide information about the location of the console backup storage in the next section of the dialog:

- Use the Storage Bucket field to specify the name of the container that will be used to store files for point-in-time recovery. This location should not change after the initial deployment of the Ark console.

- Use the Console Backup Folder field to specify a folder in which the backups will be stored.

- Use the Storage Tenant field to provide the name of the tenant in which the backup will be stored.

Use the last field to specify a timezone for the server:

- Use the drop-down listbox in the Timezone field to select the timezone that will be displayed by the Ark console.

When you're finished, click the Recover button to start the recovery process.

7.1.3 Using the Recover Option on an Azure Backed Console

If the console cannot locate a registered user, and your console is configured to support console backups, the Ark console login dialog will request the password specified during setup and display the `Deploy Console` or `Recover from Backup` options when you navigate to the console address (see Figure 7.7).

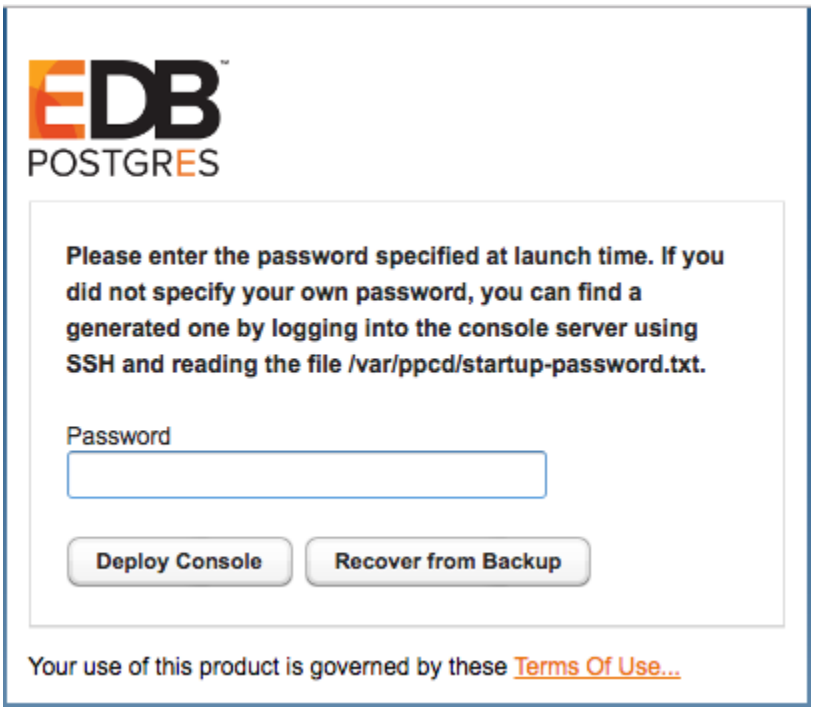

Figure 7.7 – Provide a password to Recover from Backup.

To initiate a console recovery, provide the console password specified when you deployed the console instance, and click the `Recover from Backup` button. The console properties dialog opens, prompting you for information about console backups (see Figure 7.8).

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.

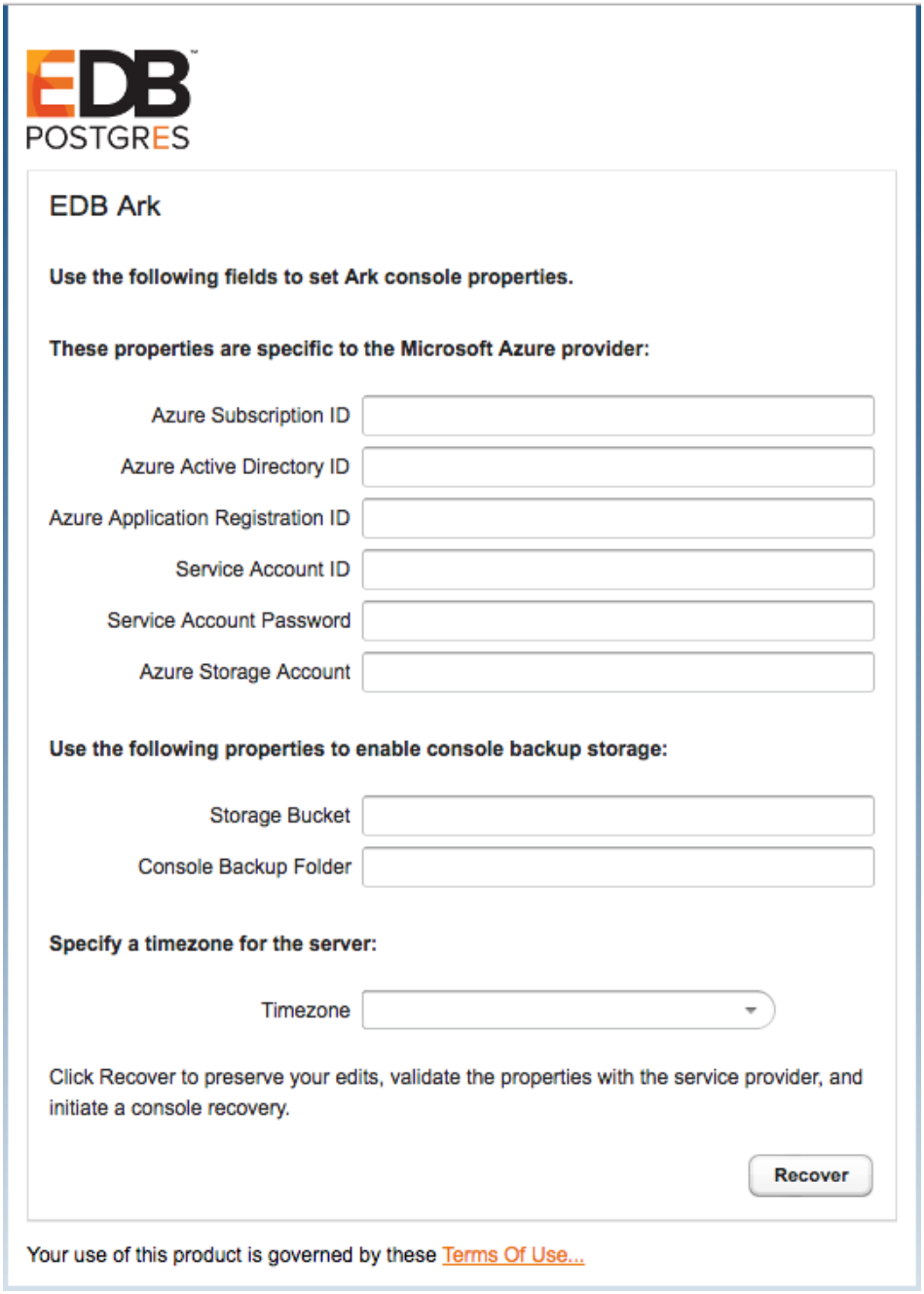

Figure 7.8 – Provide information about a console backup to recover.

Provide details about the console, and the location of a backup to recover:

- Use the `Azure Subscription ID` field to specify the subscription ID for the Azure account that hosts the Ark console.
- Use the `Azure Active Directory ID` field to specify the directory ID associated with the Azure account that hosts the Ark console.

- Use the `Azure Application Registration ID` field to specify the application ID associated with the Azure account that hosts the Ark console.

- Use the `Service Account ID` field to specify the name of the Azure service account.

- Use the `Service Account Password` field to specify the password associated with the service account.

- Use the `Azure Storage Account` field to specify the name of the Azure block storage account used with this Ark server.

Provide information about the location of the console backup storage in the next section of the dialog:

- Use the `Storage Bucket` field to specify the name of the container that will be used to store files for point-in-time recovery.

- Use the `Console Backup Folder` field to specify a folder in which the backups will be stored.

Use the last field to specify a timezone for the server:

- Use the drop-down listbox in the `Timezone` field to select the timezone that will be displayed by the Ark console.

When you're finished, click the `Recover` button to start the recovery process.

7.2 Manually Recovering from Console Backups

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.

Recovering the Console with 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. After using ssh to connect to the host of the console, you can use the following command to invoke the psql command line tool and restore the console:

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

Where *recovery_file* specifies the path and name of the backup file you wish to restore.

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

8 Notifications

EDB Ark will send e-mail notifications when:

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

Please note: For EDB Ark notifications to function properly, you must have an SMTP server running on each node, and provide contact email addresses for the Ark administrator and Ark user.

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.

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

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

	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.

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: https://enterprisedbpartners.force.com
- 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:

https://www.enterprisedb.com/products/subscriptions

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.

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
```

10 Reference

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": [
                "*"
            ]
        }
    ]
}
```

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 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 policy when registering a service role:

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

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",
```

```
"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"
}
```

```
]
}
```

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-ARK-SERVICE"
        }
      }
    }
  ]
}
```

10.5 Creating a Statically Provisioned Image

An `install.sh` script is distributed with Ark 2.3; use the script when creating a statically provisioned image. Please note: if you are creating a statically provisioned image on a RHEL host, you must register the host before configuring the cluster.

1. Create an instance that contains the backing operating system for your image.

2. Use `scp` to copy the `install.sh` file to the instance.

3. Use the following command to modify the permissions associated with the `install.sh` file:

   ```
   chmod a+x install.sh
   ```

4. Then, assume superuser privileges and invoke the `install.sh` script, including command options and values that specify details about the image:

Option	Value
`-n`	The database server type
`-v`	The database server version
`-u`	If true, Ark will invoke the yum update command and update the currently installed software packages.
`-c`	When set to true, the script will configure and enable required RHEL repositories.
`-r`	The repository address (and if applicable, credentials) for provisioning. Include the -r flag once for each repository required by packages specified with the -p or -o options..
`-p`	A list of the packages that will be installed in the image
`-o`	A list of additional packages that should be installed in the image.

5. Take a snapshot of the instance, and make the image public to make it accessible to the Ark console.

Examples

For example, the following command creates a static image that contains the EDB Postgres Advanced Server 9.6 database on a RHEL host:

```
$ sudo ./install.sh -n ppas -v 9.6 -u true -c true \
-r http://USERNAME:PASSWORD@yum.enterprisedb.com/9.6/redhat/rhel-
\$releasever-\$basearch \
-r http://USERNAME:PASSWORD@yum.enterprisedb.com/tools/redhat/rhel-
\$releasever-\$basearch \
-r
http://USERNAME:PASSWORD@yum.enterprisedb.com/dependencies/redhat/rhel-
\$releasever-\$basearch \
-p "edb-as96-server edb-pgpool35 edb-as96-pgpool35-extensions" \
```

The following command creates a static image that contains PostgreSQL 9.6 on a CentOS host:

```
$ sudo ./install.sh -n postgres -v 9.6 -u true -c false \
-r http://yum.postgresql.org/9.6/redhat/rhel-7-x86_64/pgdg-redhat96-
9.6-3.noarch.rpm \
-p "postgresql96-server pgpool-II-96" \
```

Please note: the backslash must be the last character on each of the above lines (whitespace may not follow the backslash character).

> The script returns `Script execution complete` when the command finishes executing successfully.

When creating a new server with the Ark console that references the image, check the box next to `Statically Provisioned` on the console properties dialog. For more information about defining a server, see Section 4.1.2 .