EDB™ Ark

Administrative User's Guide

Version 3.3

April 17, 2019

EDB Ark Administrative User's Guide, Version 3.3
by EnterpriseDB® Corporation

EnterpriseDB Corporation, 34 Crosby Drive, Suite 201, Bedford, MA 01730, USA
T +1 781 357 3390 **F** +1 978 467 1307 **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. 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:

- Monitoring
- 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 provides unattended, around-the-clock responsiveness to unpredictable load demands on your database infrastructure.

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

- **EDB Ark - Overview** – Section 2 provides information about EDB Ark functionality and architecture, using Ark with a PEM server, and the available Ark Authentication models.

- **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.

- **Reference** – Section 10 provides reference information about the AWS policies required by the Ark console.

- **Creating a Statically Provisioned Instance** - Section 11 provides detailed information about how to create a statically provisioned instance. . A statically provisioned server is a pre-configured image that contains the software required to create a database cluster.

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 the EDB Ark API for release 3.3:

- A call to the `/options/roleinfos` resource returns a list of AWS externalId/roleArn pairs; for more information, see the *EDB Ark API User's Guide*.

- A call to the `/options/vpcids/{tenant}?usePrivateIps=[true | false]` resource returns information about private IP support; for more information, see the *EDB Ark API User's Guide.*

1.2 Typographical Conventions Used in this Guide

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

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

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

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 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.

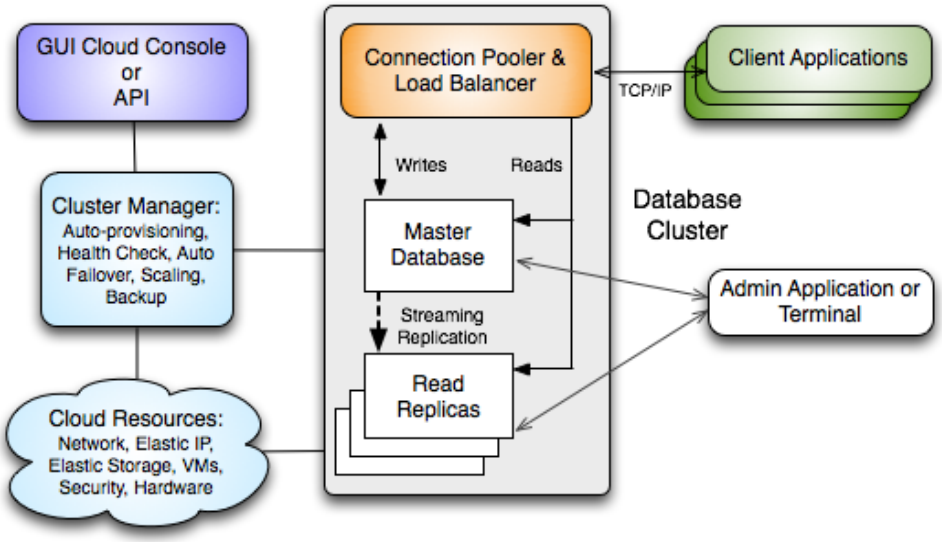

Figure 2.1 - An overview of the EDB Ark architecture.

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.

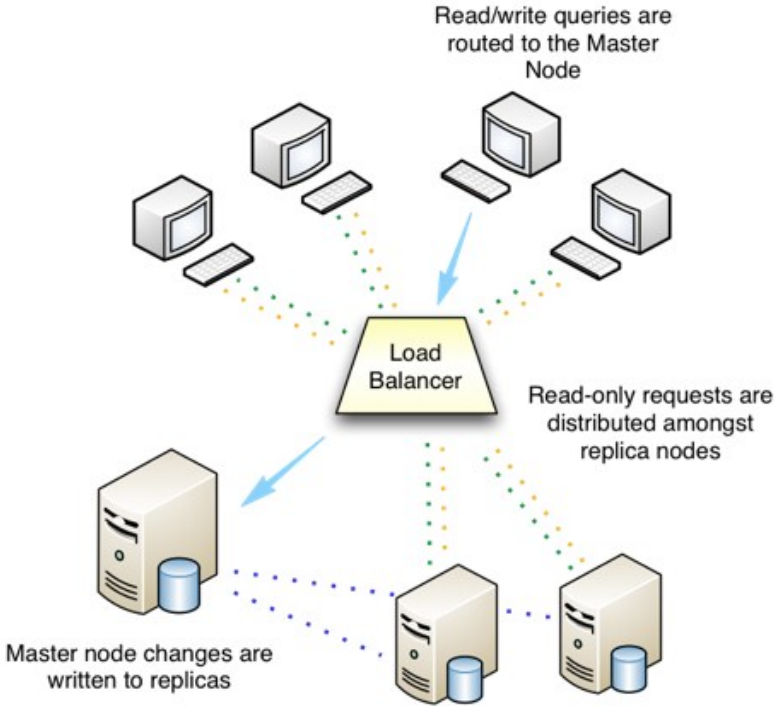

Figure 2.2 - EDB Ark performs automatic load balancing.

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 Ark on an Amazon AWS Virtual Private Cloud

EDB Ark can create and manage clusters that reside on Amazon-hosted virtual private clouds (VPCs). A 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 addresses. When deploying a cluster, you can use the Ark console to select the VPC on which the new cluster will reside, or choose to have Ark create a new VPC.

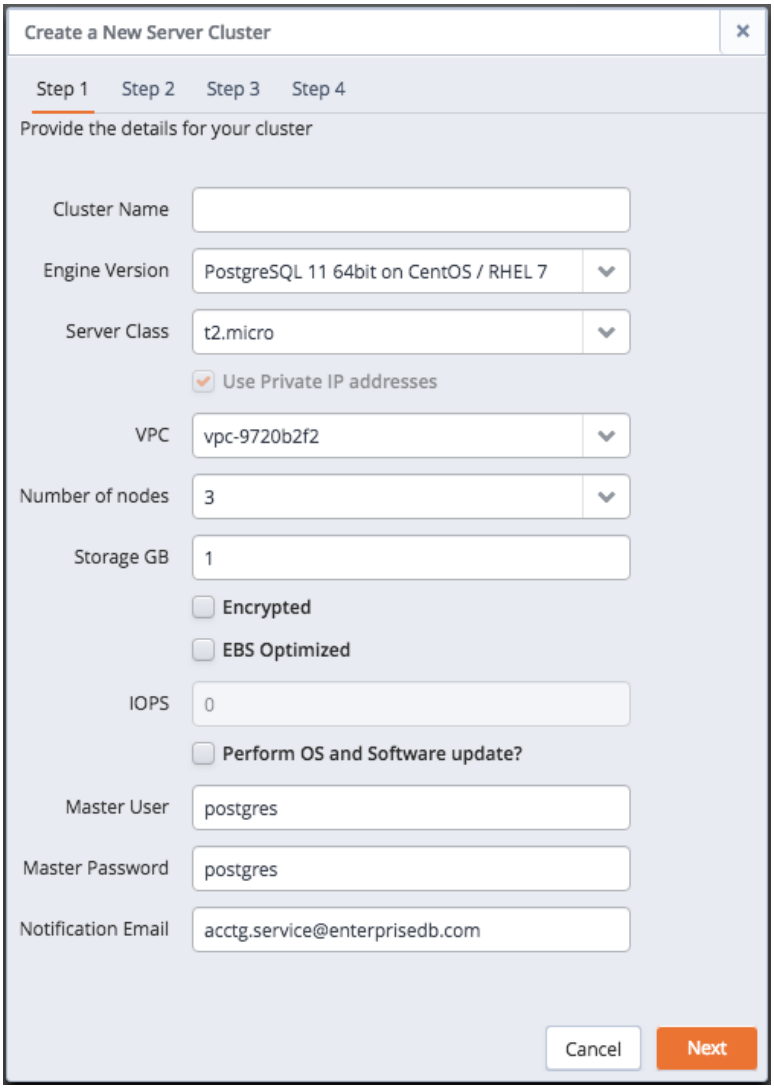

Figure 2.5 - Creating a new Ark cluster - the Step 1 tab.

To create a new cluster that resides on a private subnet, log into the Ark console and click the `Launch DB Cluster` button. Use the `Create a new Server Cluster` dialog

(see Figure 2.5) to provide details about the cluster configuration. Check the box to the left of `Use Private IP addresses` to display only those VPCs which have a NAT gateway configured to support private subnets in the `VPC` field. Then, use the `VPC` drop-down menu to select a VPC.

After completing the `Step 1` tab, use the `Next` key to continue. Provide information in the fields on each additional tab before selecting the `Launch` button and deploying your cluster into your private subnet.

For detailed information about the additional options available when defining a cluster, please see the *EDB Ark Getting Started Guide*, available via the Ark console dashboard.

Please note: if you use private IP addresses, the master instance is not assigned an elastic IP address. Should a failover occur, the IP address of the master instance will change.

Using a NAT Gateway

You can deploy the Ark console on a VPC, and use a network address translation (NAT) gateway to provide access to services outside of the VPC. The NAT gateway allows an instance without a public IP address to securely access services and resources such as yum repositories. For more information about using a NAT gateway, visit:

https://docs.aws.amazon.com/AmazonVPC/latest/UserGuide/vpc-nat-gateway.html

When the Ark console is deployed in a private subnet (or without a public IP address), the console can only communicate with private networks in its own VPC or peered VPCs. Clusters are restricted to deploying into VPCs that have a peering connection to the VPC in which the console is deployed, and the console's VPC.

A peering connection allows you to route traffic between two virtual private clouds without exposing the clouds to outside connections. For detailed information about using peering, visit:

https://docs.aws.amazon.com/AmazonVPC/latest/PeeringGuide/Welcome.html

Please note: when the Ark console is deployed in a private subnet, the `Use Private IP addresses` option is always `true`.

2.2 Registering an Ark Cluster with Postgres Enterprise Manager

Postgres Enterprise Manager (PEM) is an enterprise management tool designed to assist database administrators, system architects, and performance analysts in administering, monitoring, and tuning PostgreSQL and EnterpriseDB Advanced Server database servers. PEM can manage and monitor a handful of servers or hundreds of servers from a single console, allowing complete control over all aspects of your databases.

A PEM installation consists of a PEM server, one or more PEM agents, and the backing database server (named `pem`). The PEM server includes a web interface that allows you to monitor and manage the database instances that are registered with a PEM agent. A PEM agent is responsible for returning metric information to the PEM server, and performing tasks on the database instances that are registered with that agent.

The Ark console installation includes a pre-configured PEM server, a PEM agent, and the `pem` backing database. You also have the option of using a *remote* PEM server to monitor your Ark console. A remote server is a PEM server that has been installed and configured on another host.

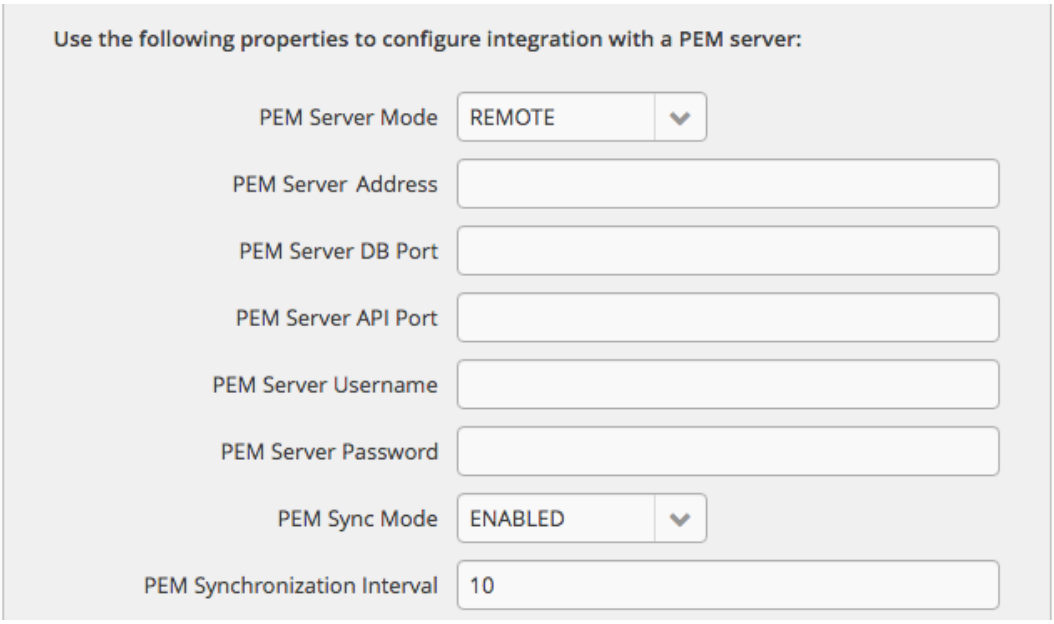

Figure 2.8 - the PEM Server configuration properties.

During deployment, you will be prompted to use the `PEM Server Mode` drop-down listbox to select a deployment mode:

- Select `DISABLE` to indicate that clusters deployed on the host should not be registered with the PEM server.

- Select `LOCAL` to indicate that you would like to use the PEM server that resides on your local host. If you select `LOCAL`, the PEM deployment will use default values assigned by the installer.

    - The IP address of the PEM server host will be the IP address of the Ark host.
    - The `PEM Server DB Port` will monitor port `5432`.
    - The PEM server database user will be named `postgres`.
    - The password associated with the PEM server will be the same password as the Ark console.

- Select `REMOTE` to indicate that you would like to use a PEM server that resides on another host, and provide connection information on the Ark console deployment dialog (see Figure 2.8).

    - Provide the host name or IP address of the PEM server host in the `PEM Server Address` field.
    - Specify the port monitored for connections by the PEM server in the `PEM Server DB Port` field.
    - Specify the port monitored for API connections in the `PEM Server API Port` field.
    - Provide the name that should be used when authenticating with the PEM server in the `PEM Server Username` field.
    - Provide the password associated with the PEM server user in the `PEM Server Password` field.

If you select `REMOTE`, whenever a new cluster node is created on this console, it will be registered for monitoring by the PEM server. Please note that you must modify the `pg_hba.conf` file of the `pem` database on the remote PEM server to accept connections from the host of the Ark console. For detailed information about modifying the `pg_hba.conf` file, please see:

https://www.postgresql.org/docs/10/static/auth-pg-hba-conf.html

2.2.1 Syncing with the PEM Server

When you register with PEM during console deployment, the `PEM Sync Mode` field will be enabled. Use the drop-down listbox in the `PEM Sync Mode` field to specify your preference for synchronizing with the PEM server.

If you select `DISABLED`:

- Any object that Ark registers with PEM will be owned by the PEM user specified in the `PEM Server Username` field (if the PEM server is a `REMOTE` server), or by `postgres` (if the PEM server resides on the host of the `LOCAL` Ark server).

- Any Ark user that registers with a console instance will *not* be added as a PEM user.

If you select `ENABLED`:

- Any Amazon Role or Azure Group that is accessible by the Ark user that is deploying the console may access is added to the PEM server as a group role. The PEM role created for the project, role, or group is not a login role.

  To simplify management, each role that represents a project, role, or group is also a member of the PEM `ark-team` group role (see Figure 2.9).

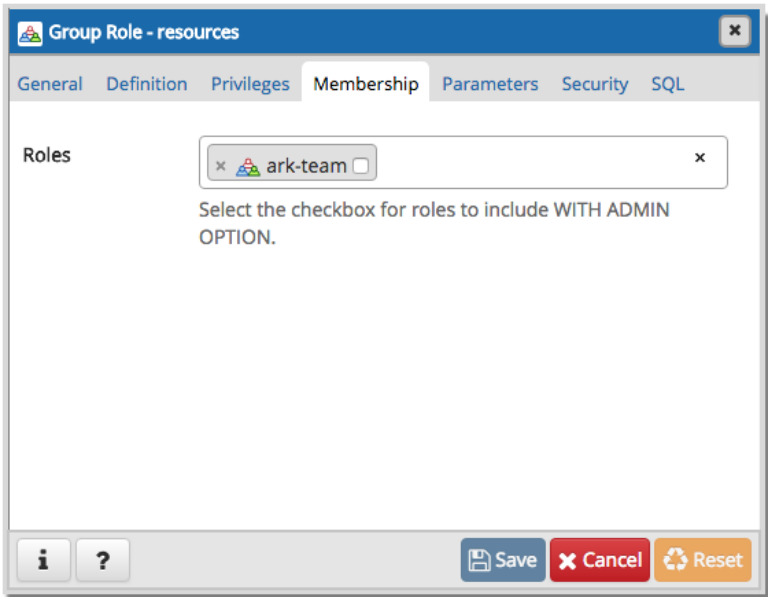

Figure 2.9- The Membership tab of the PEM web interface Group Role dialog.

- Each user that registers with a monitored Ark instance is also created as a PEM Login user. The user account will be displayed in the `Login/Group Roles` node of the PEM client tree control. To access the properties dialog for the user, right click on the user name, and select `Properties` from the context menu.

  Each PEM user account that corresponds with a registered Ark user will belong to the `ark-user` role. The PEM user account will also have membership in any `ark-team` roles that have been created to correspond to the project, role, or groups that the Ark user has permission to access (see Figure 2.10).

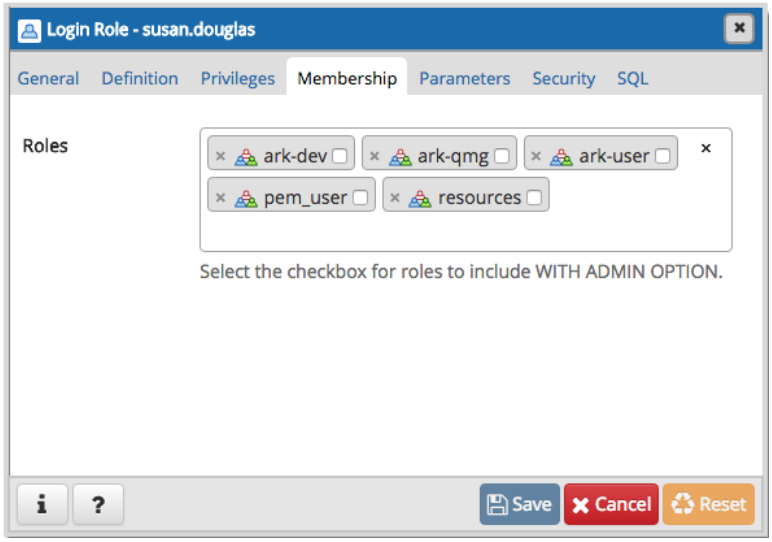

Figure 2.10 - the Membership tab of the PEM web interface Login Role dialog.

The PEM user account is not password enabled. To set a password for the account, an administrative user must navigate to the PEM role's `Definition` tab, and provide a password in the `Password` field (see Figure 2.11).

If you use the Ark console to delete an Ark user, the synchronization service will disable the corresponding user on the PEM server. The service will ensure that membership in `ark-user` is revoked, and the role will no longer be a `LOGIN` role. Please note that the sync service will only modify those roles that are a member of the `ark-user` team.

The sync service will also ensure that the corresponding `pem-team` role is deleted if a tenant, group, or role is deleted on an Ark host.

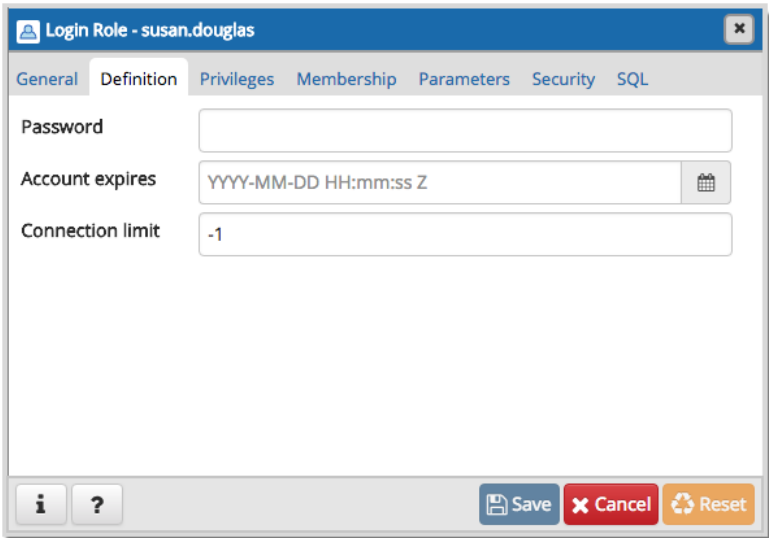

Figure 2.11 - the Definition tab of the PEM web interface Login Role dialog.

Ark will sync with PEM on the schedule specified by the `PEM Synchronization Interval` field. The field accepts interval values in minutes; by default, Ark will attempt to syncronize every 10 minutes.

2.2.2 Monitoring an Ark Cluster

Accessing the PEM Web Interface

After deployment, you can access the PEM web interface in your choice of browser. If the PEM server resides on a LOCAL host, navigate to:

```
https://address_of_ark_server/pem
```

If the PEM server resides on a REMOTE host, navigate to:

```
https://address_of_remote_pem_server:port/pem
```

When prompted (see Figure 2.12), provide the PEM credentials to connect.

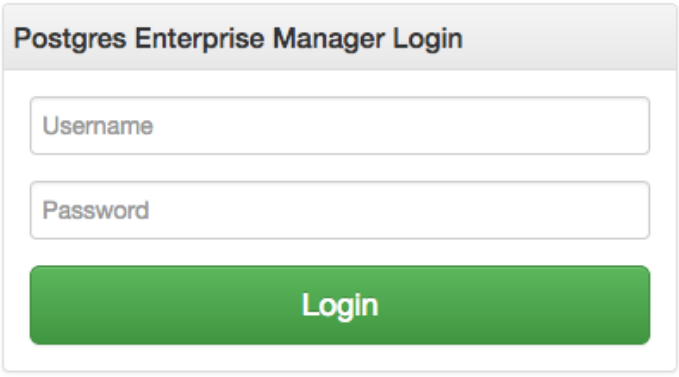

Figure 2.12 - the PEM Server prompts you for credentials.

If you have registered the Ark console with a remote PEM server during deployment, use the PEM server credentials to connect.

If you have deployed the PEM server locally (on the Ark host), the password associated with the Ark backing database will be used for the PEM server. Unless you have modified the password (either during deployment in the `DB User New Password` field, or after deployment in the console backing database), the Ark database superuser has the following connection credentials at deployment:

name: `postgres`
password: `0f42d1934a1a19f3d25d6288f2a3272c6143fc5d`

You should change the password after deployment to a unique password (known only to trusted users). For details, see section 6.2.

After authenticating with the server, the PEM web interface allows you to manage your monitored nodes (see Figure 2.13).

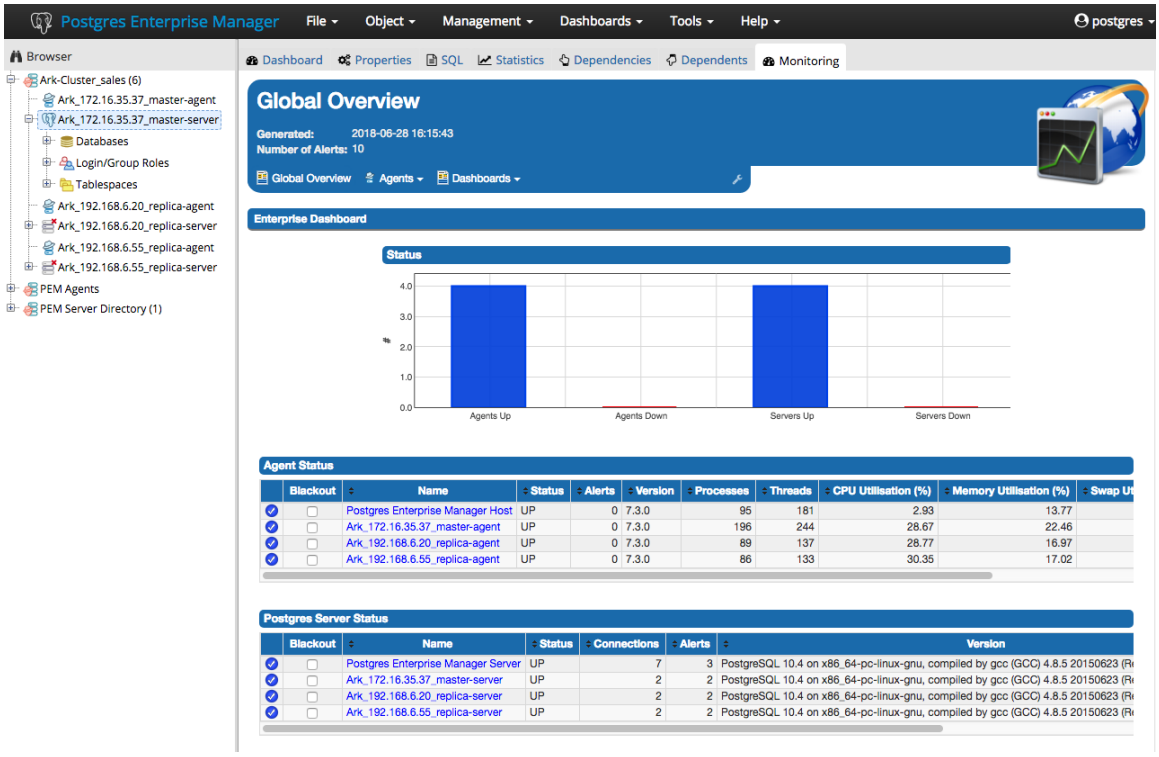

Figure 2.13 - The PEM console, displaying Ark console members.

When you launch an Ark cluster on a console that is registered with PEM, Ark will register each node of the cluster with the PEM server for monitoring. Each node of an Ark cluster, and the agent that resides on the node is displayed in a Group-level heading in the PEM web interface `Browser` tree control. The node name is:

```
Ark-Cluster_cluster_name
```

Where *`cluster_name`* is the name assigned to the cluster.

Right click on the IP address of a cluster node, and provide the password specified when the cluster was provisioned to authenticate with the database and view the database objects that reside on the cluster in the tree control (see Figure 2.12).

PEM documentation is available via the PEM web interface `Help` menu, or at the EnterpriseDB website at:

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

Known Limitations

If your Ark clusters are provisioned with private IP addresses, they may not be reachable from the PEM server. If this is the case, you will not be able to use the PEM Dashboard to remotely browse the database server. PEM agents running on the Ark cluster nodes will be able to report status to the PEM Server.

Please note that the user identifier associated with an Ark cluster (the cluster owner) must be unique across all Ark consoles supported by a given PEM server.

2.2.3 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.

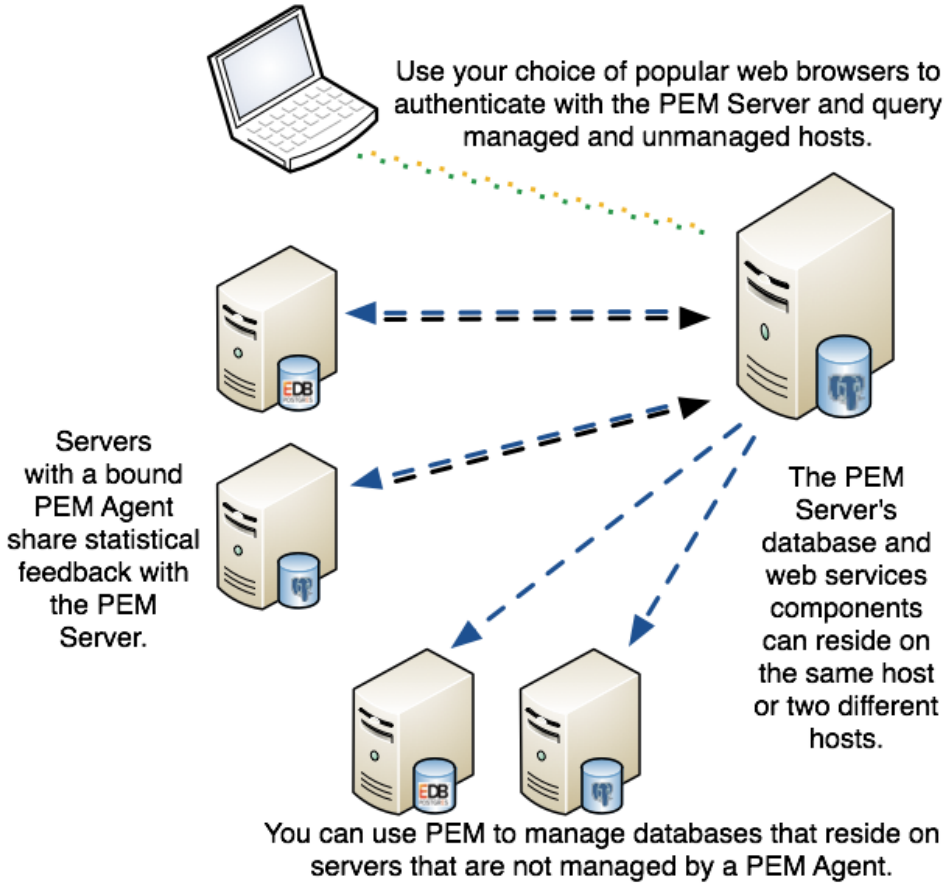

Figure 4.21 – A typical PEM installation.

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.

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

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.

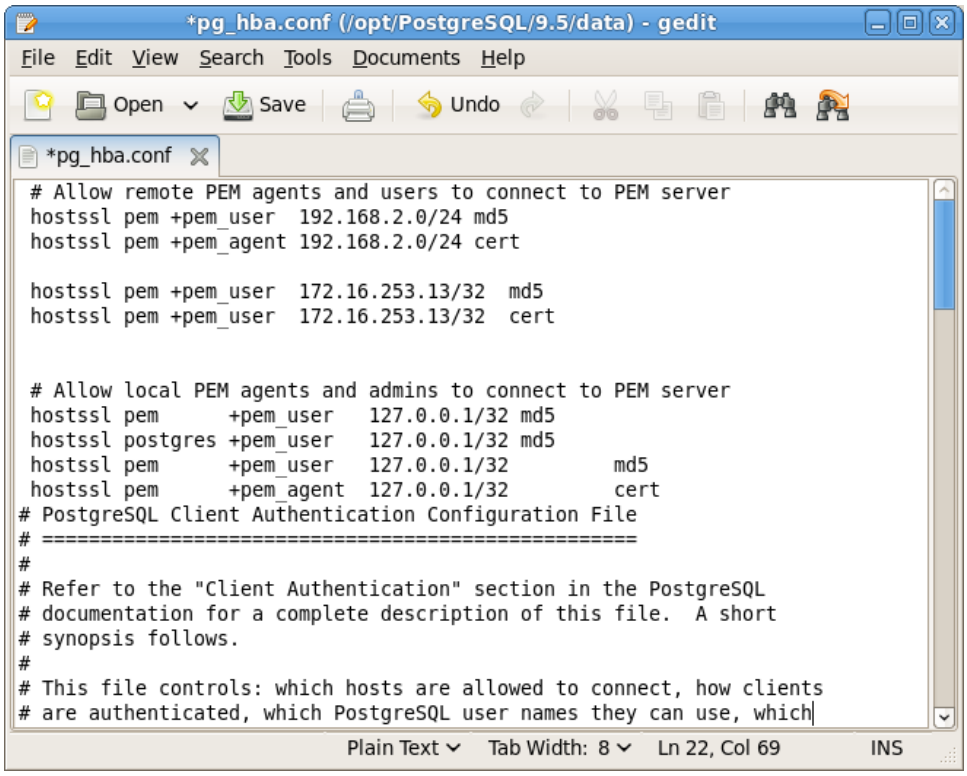

Figure 4.24 – Modifying the PEM Server's pg_hba.conf file.

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-x
```

Where *x* specifies the version. For example:

```
service postgresql-10 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).

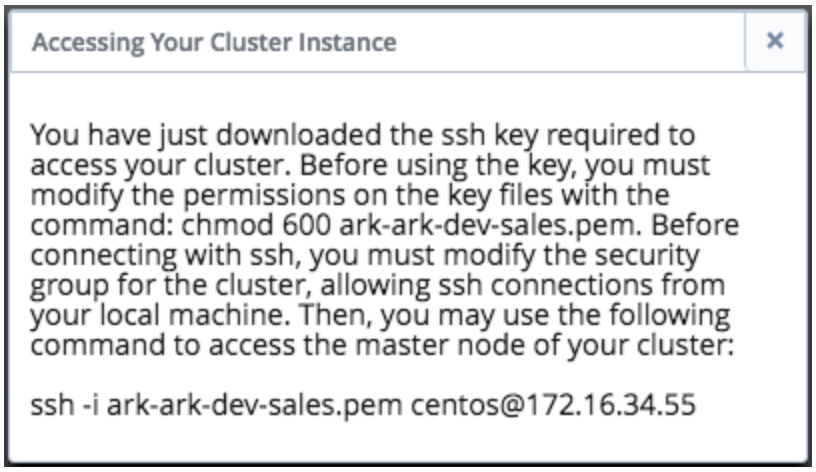

Figure 4.25 - Using SSH to connect to the Ark cluster.

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).

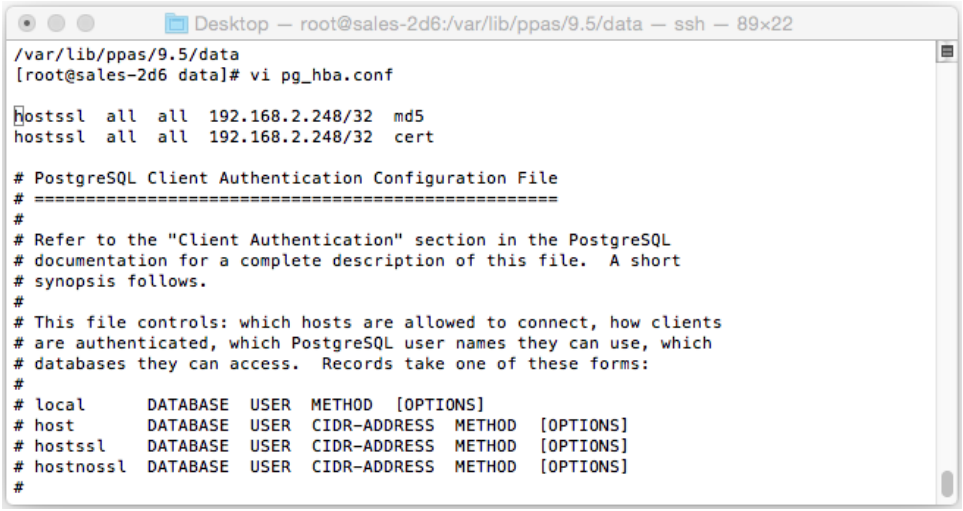

Figure 4.26 – Modifying the Ark cluster's pg_hba.conf file.

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

2.3 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 Azure: use the 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: 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.

2.3.1 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.

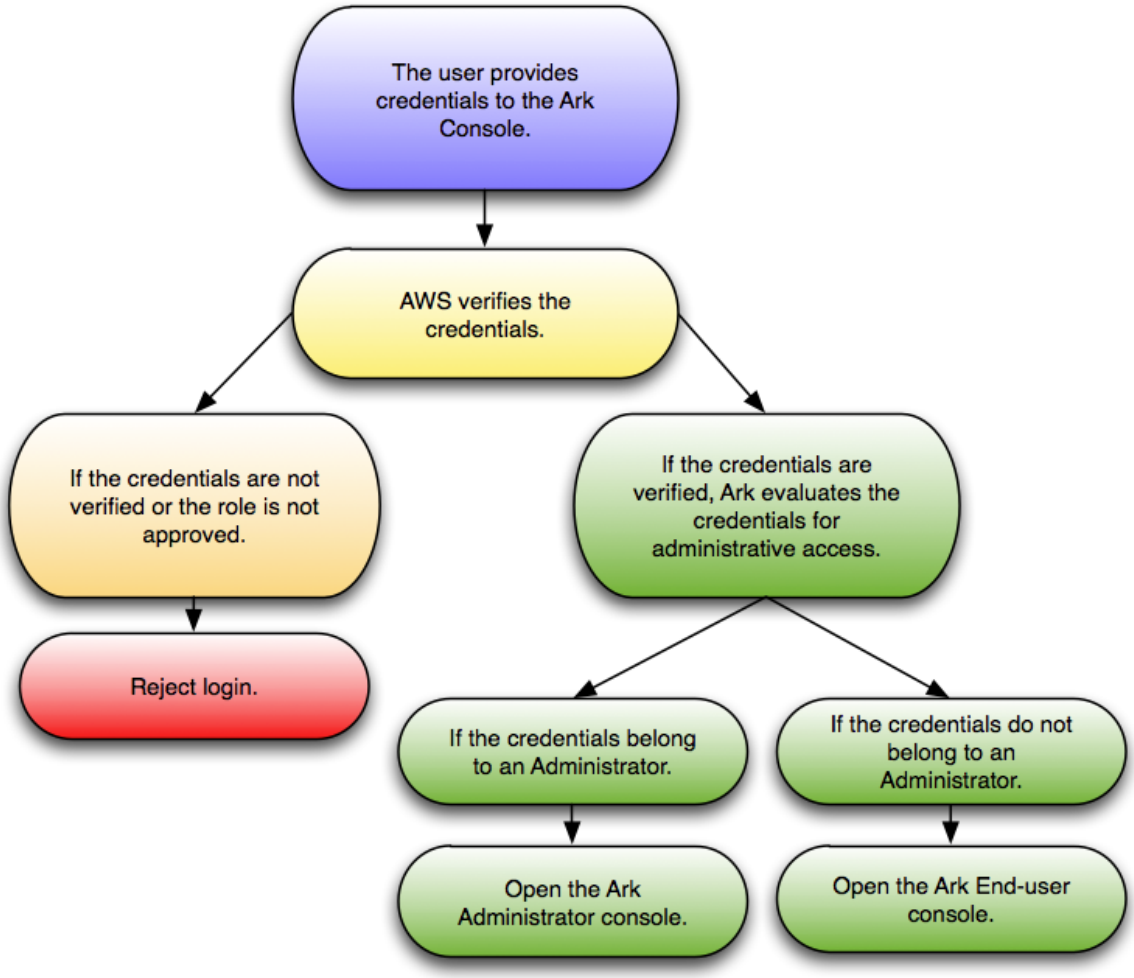

Figure 2.11 - Using provider authentication on Amazon.

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.

2.3.2 Using PostgreSQL Authentication on AWS

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

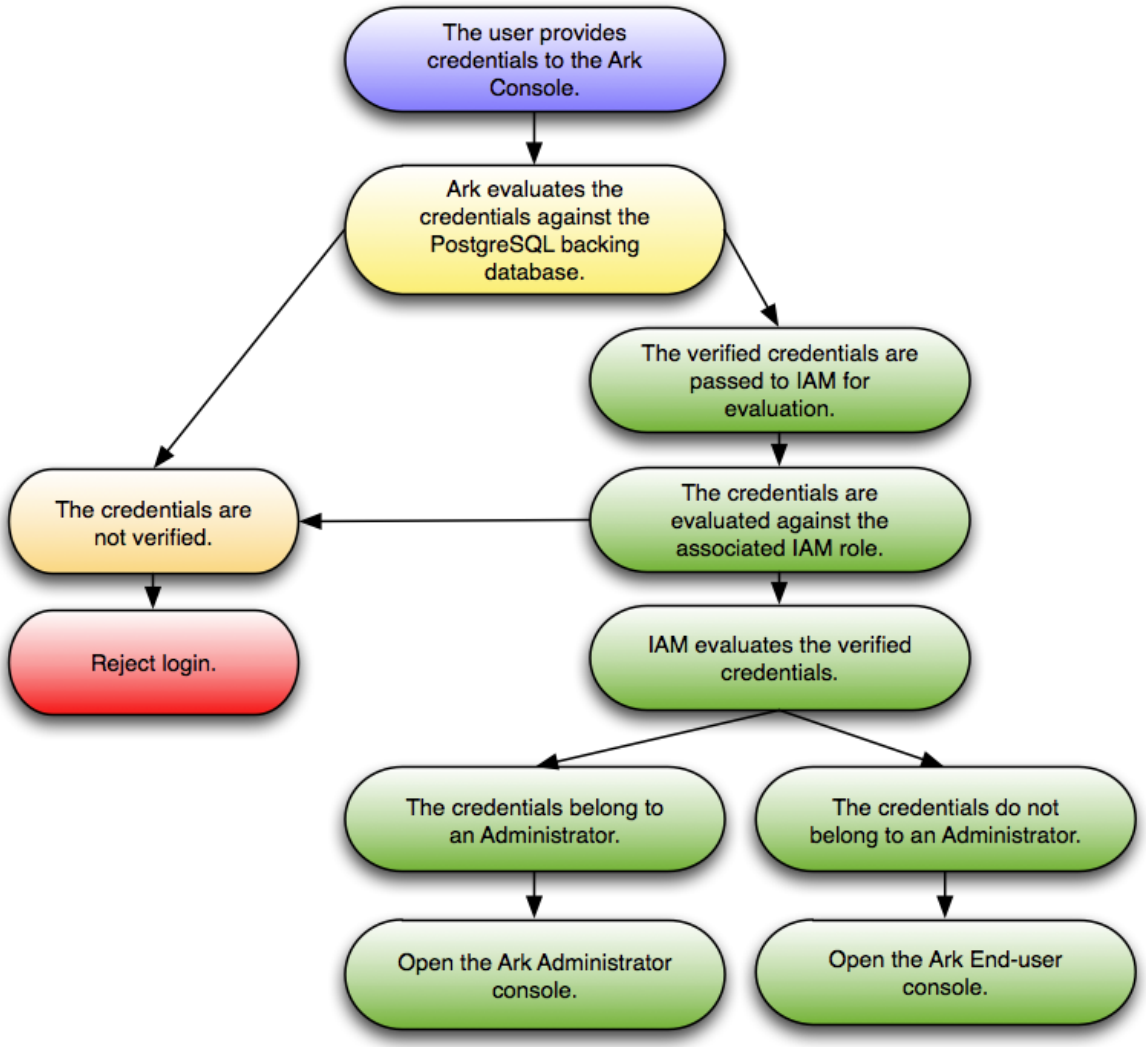

Figure 2.12 - Using Postgres authentication on AWS.

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 2.3.

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

2.3.3 Using Provider Authentication on Azure

If you use native password authentication provided by Azure:

- You must use the Azure Active Directory console to create and manage user accounts.

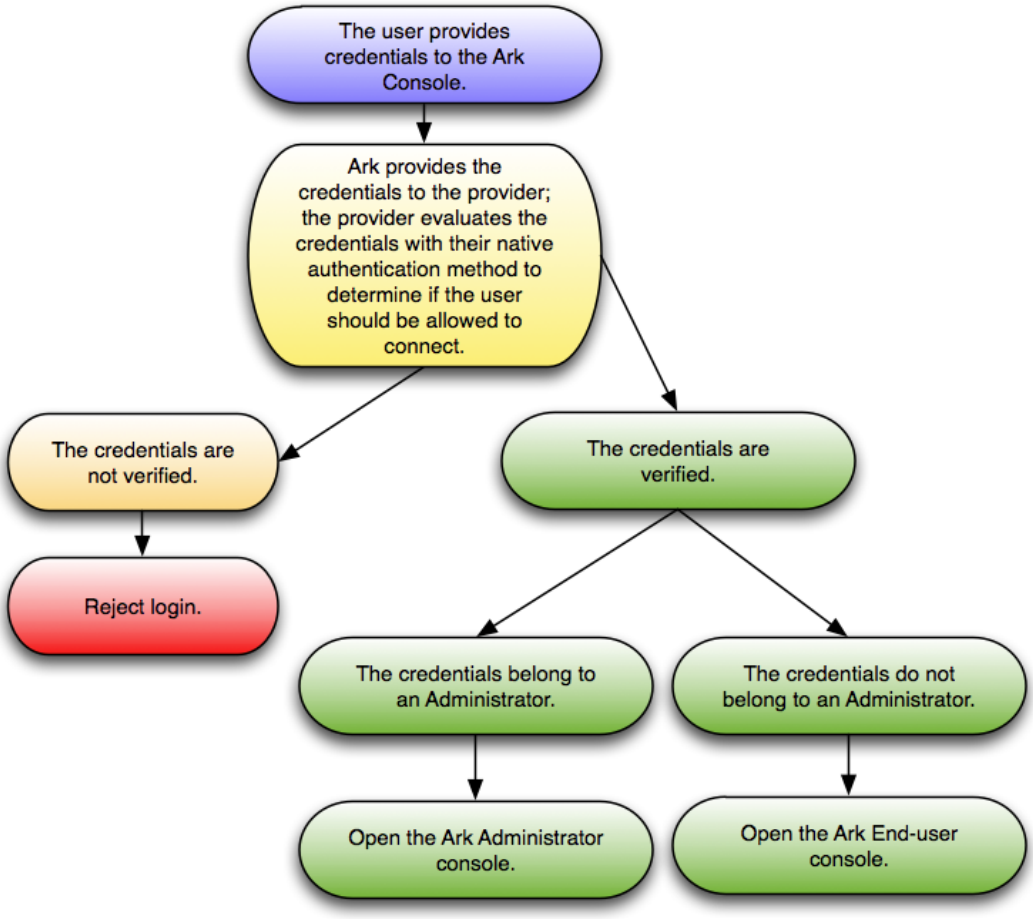

Figure 2.13 - Using provider authentication on Azure.

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.

2.3.4 Using PostgreSQL Authentication on Azure

When Postgres authentication is enabled on 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.

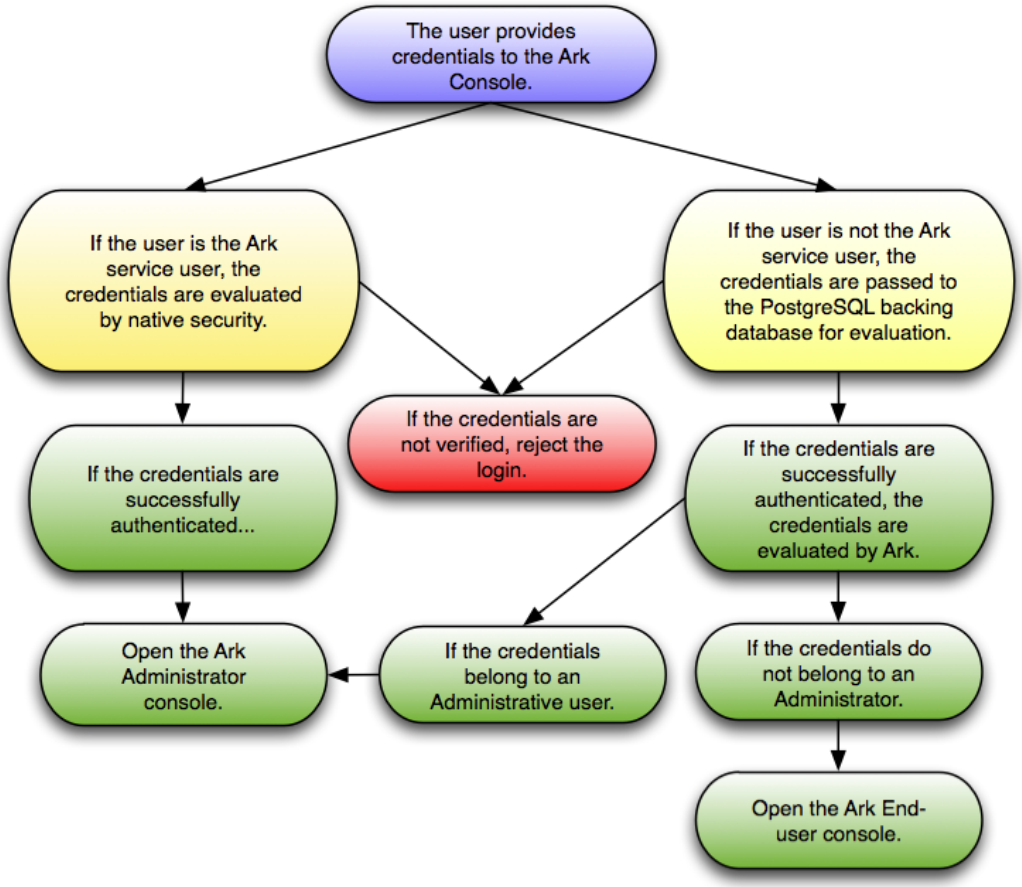

Figure 2.14 - Using Postgres authentication on Azure.

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.

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 Azure host, see Section 3.2 for detailed console installation information.

3.1 Installing EDB Ark for Amazon AWS

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

1. 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.

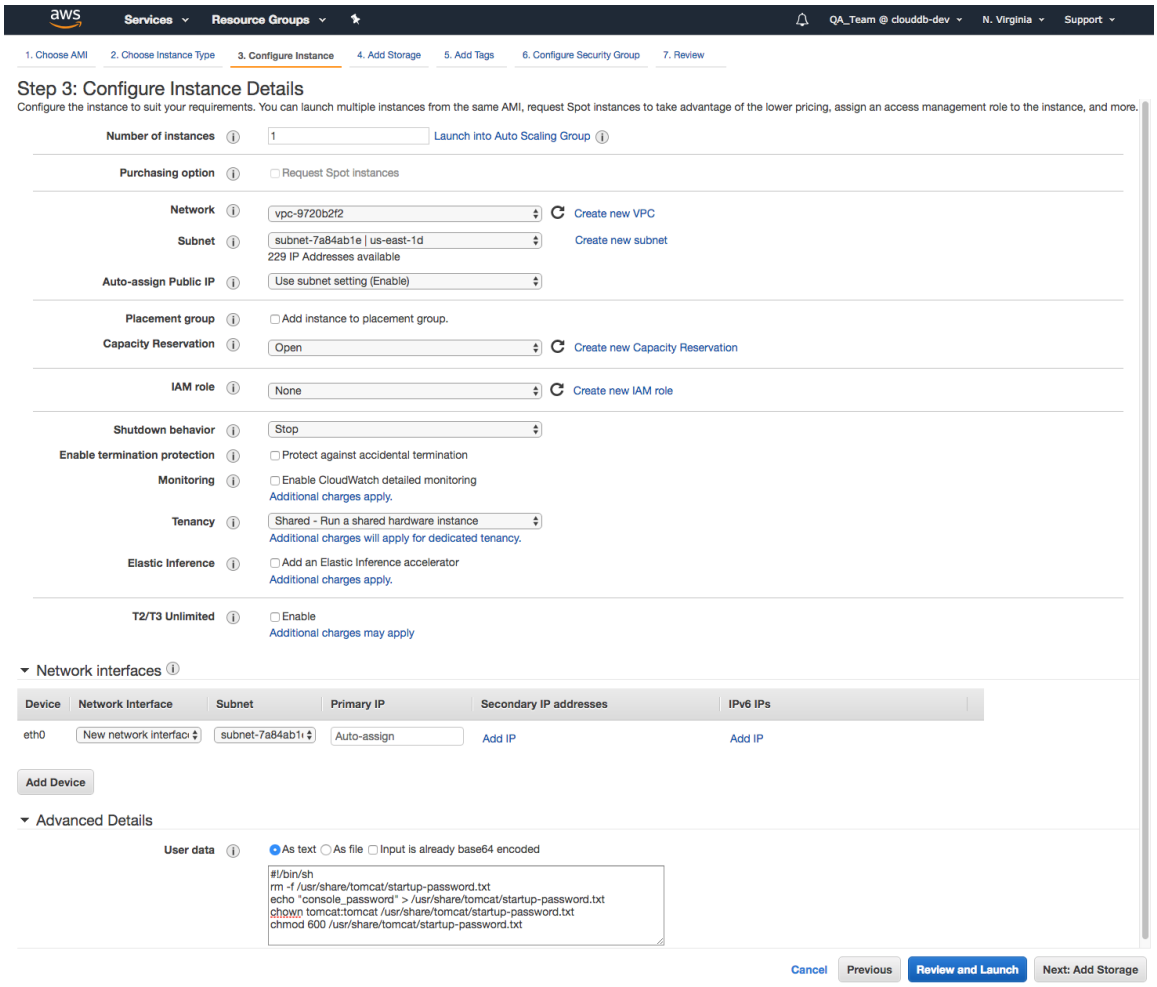

Figure 3.1 – Step 3 - Enabling the startup script.

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 /usr/share/tomcat/startup-password.txt
echo "console_password" > /usr/share/tomcat/startup-
password.txt
chown tomcat:tomcat /usr/share/tomcat/startup-password.txt
chmod 600 /usr/share/tomcat/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.

Continue through the launch wizard; please note that when configuring your security group, the group must allow communication between the nodes of the cluster.

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
Custom TCP	Ingress	5432	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 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 through the `IAM` menu (Identity and Access Management) to the `Users` dashboard; select the `Add user` button to open the `Add user` dialog (shown in Figure 3.2).

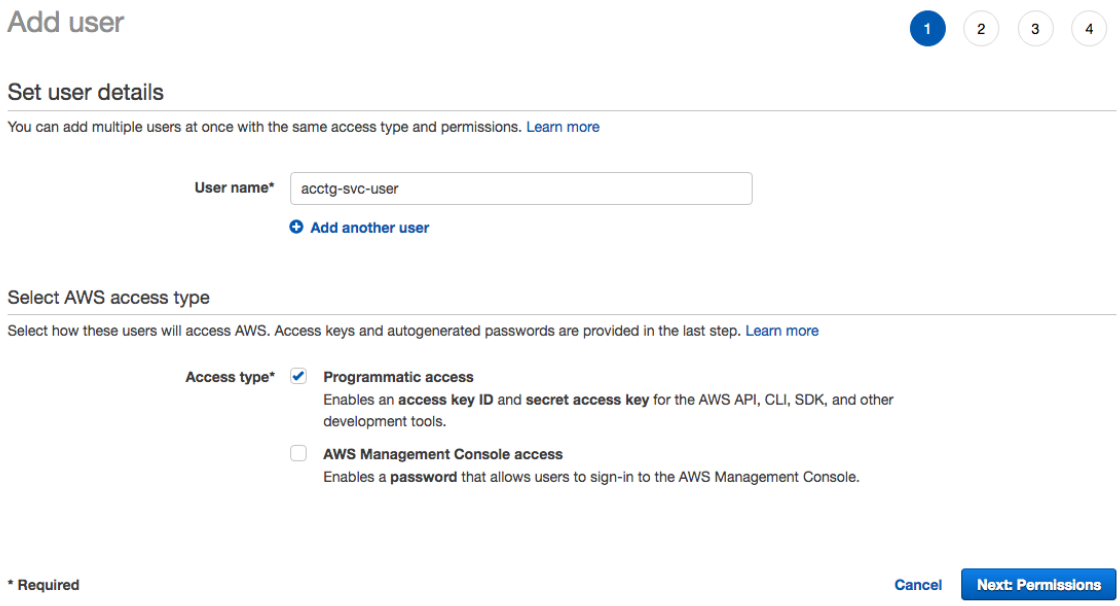

Figure 3.2 - The Add user dialog.

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. Click the `Attach existing policies directly` button, and then the `Create policy` button to open the `Create policy` dialog in a new tab.

When the `Create policy` dialog opens, select the `JSON` tab, and provide the policy definition (see Figure 3.3).

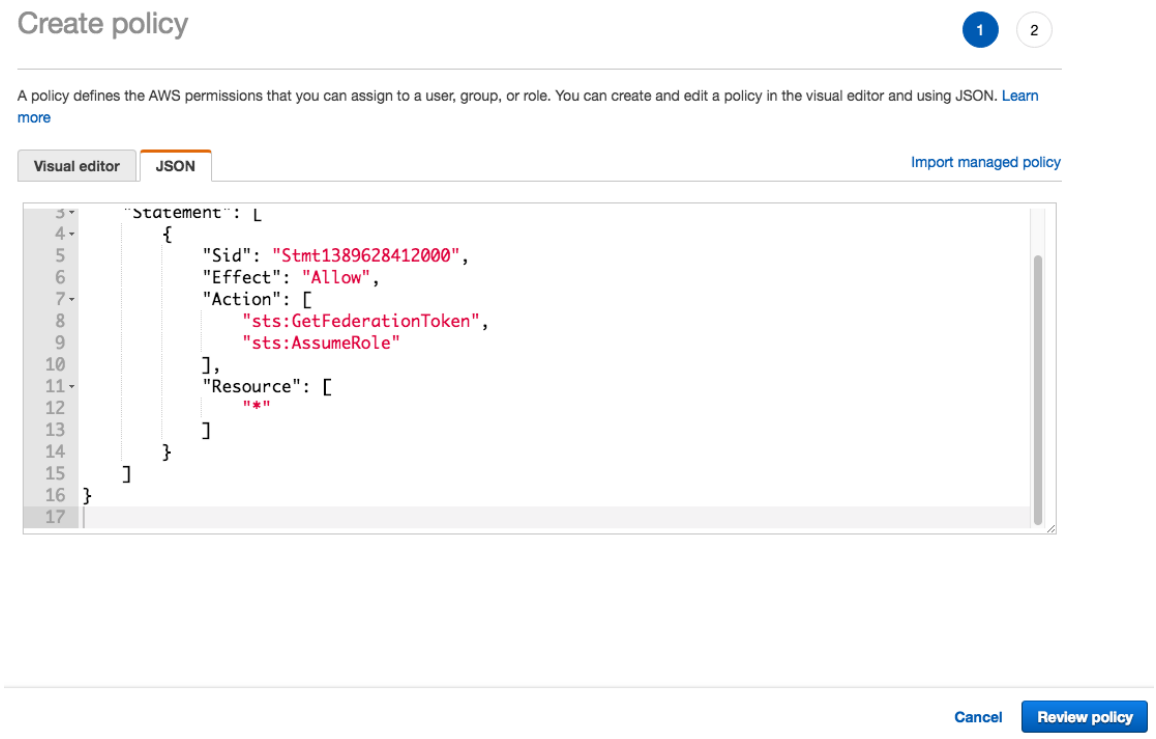

Figure 3.3 - Adding a policy definition.

A sample policy definition is available in Section 10.1; after copying in a policy, click the `Review policy` button to continue.

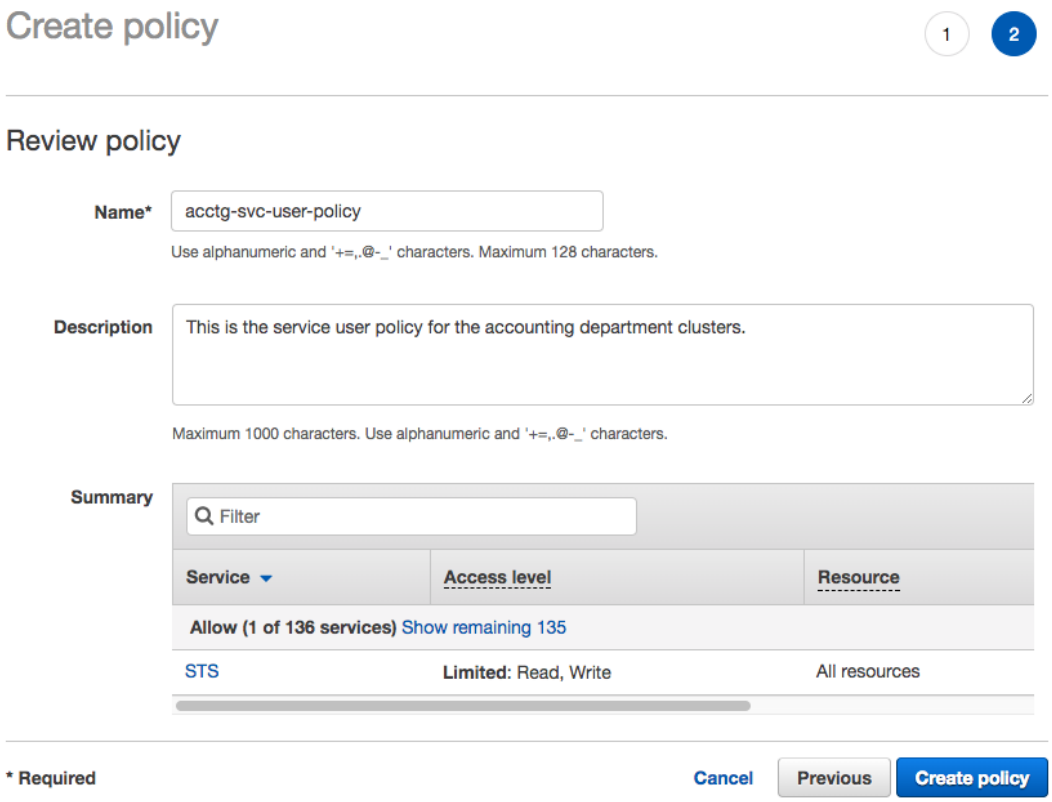

Figure 3.4 - Completing the policy definition.

Provide a name and a description for the service policy definition (see Figure 3.4), and click the `Create policy` button to continue.

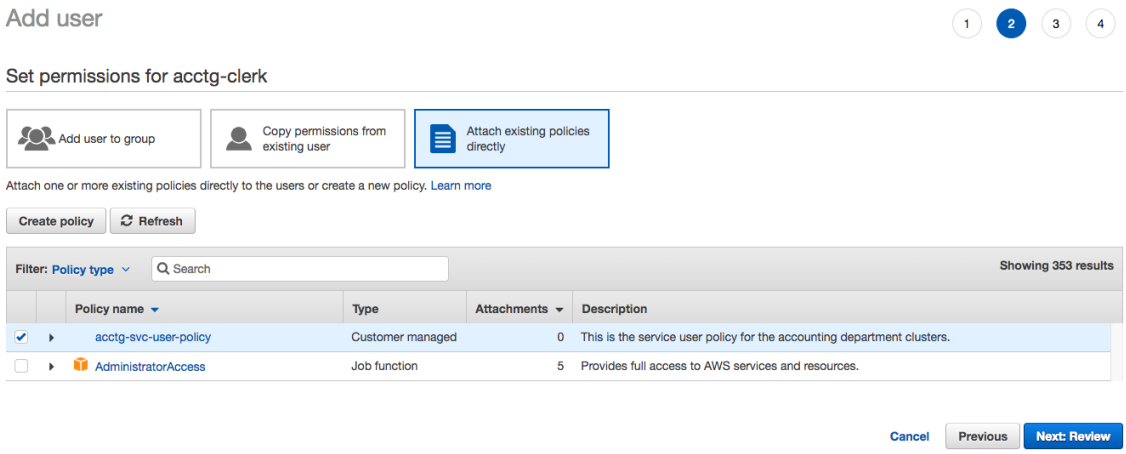

Figure 3.5 – Attaching the policy.

Return to the `Add user` tab, and click the `Refresh` button. Check the box to the left of the new policy, and click `Next: Tags` (see Figure 3.5).

IAM user tags are optional; you can click `Next: Review` to continue.

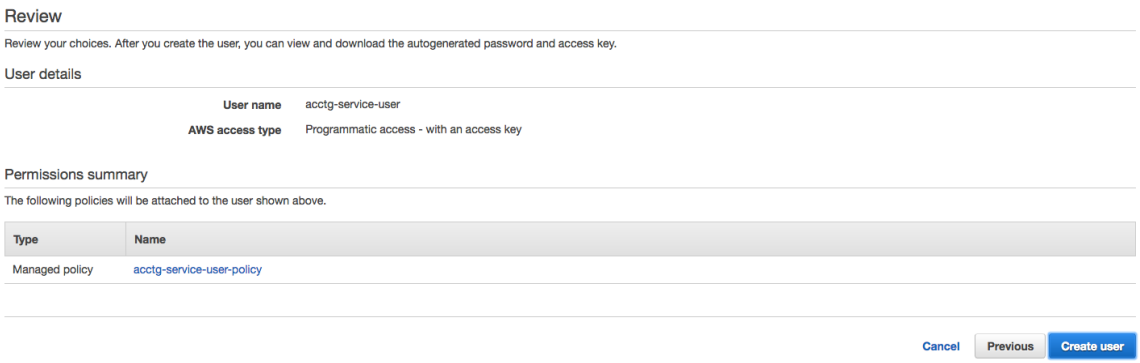

Figure 3.6 – Creating the user.

Review the account details, and click the `Create user` button to create the user (see Figure 3.6). The AWS console will confirm that the user has been added successfully (see Figure 3.7).

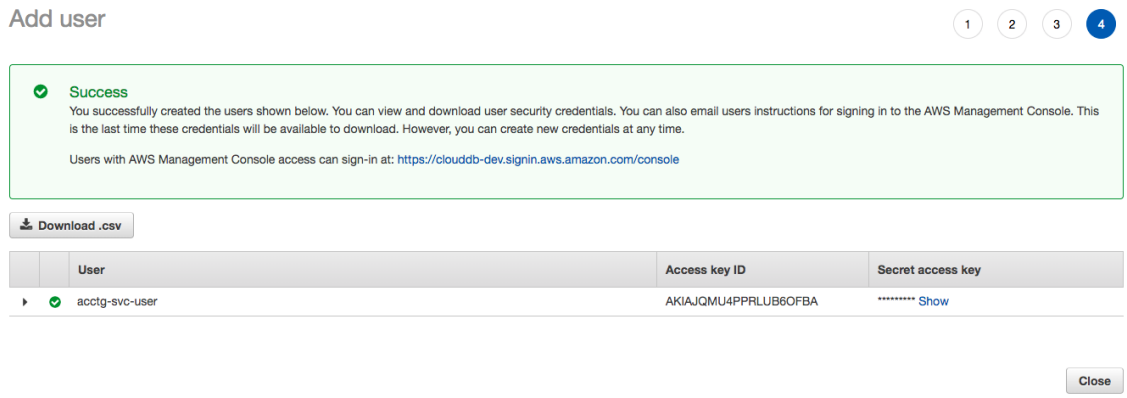

Figure 3.7 – The user is created successfully.

Keep a copy of 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.
- Use the `Show` button to display the `Secret access key`. You must 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. Connect to the Amazon management console, and navigate through the `Identity and Access Management` dashboard to the `Roles` dashboard. Then, click the `Create role` button to open the `Create role` dialog (see Figure 3.8).

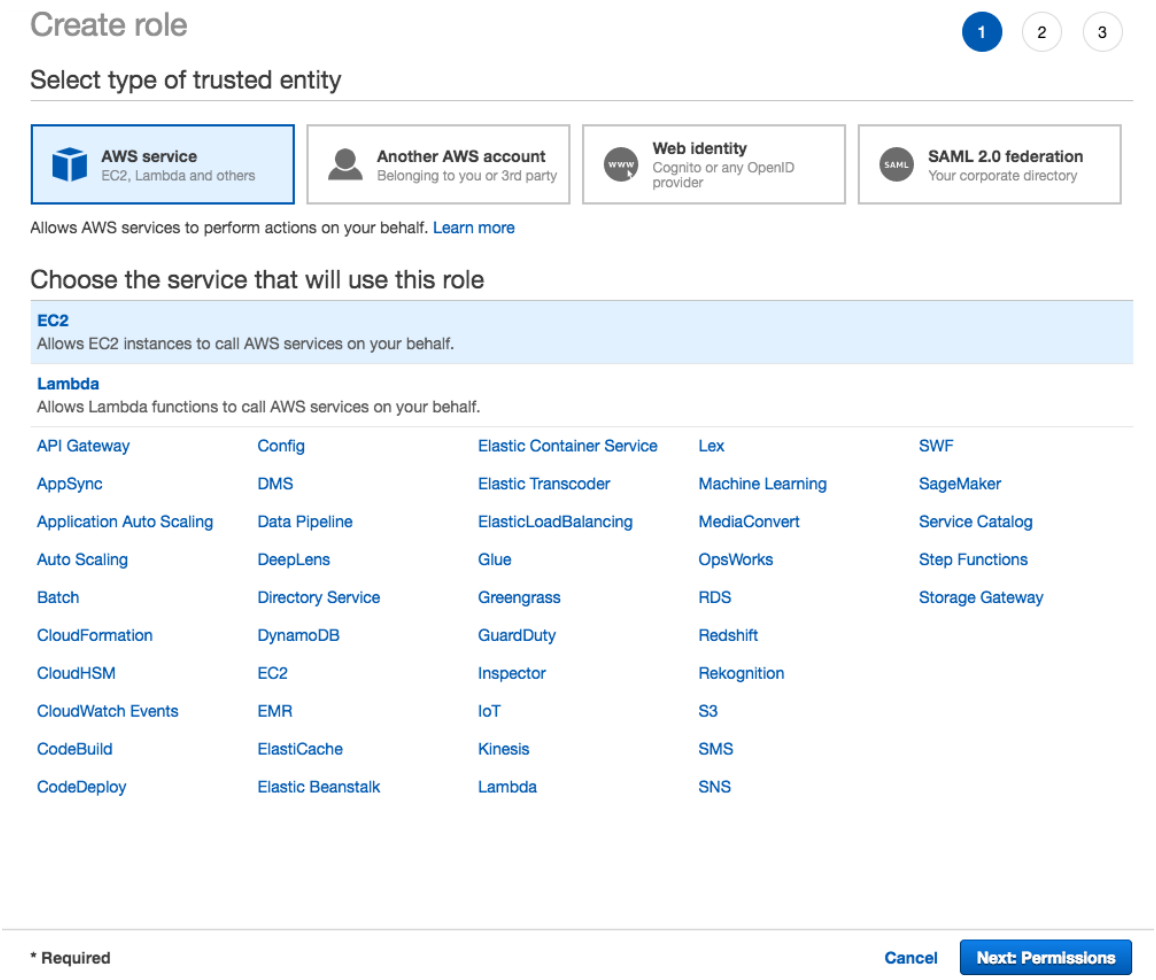

Figure 3.8 - Creating a role.

Select the `AWS service` button, and the `EC2` service type; click `Next: Permissions` to continue.

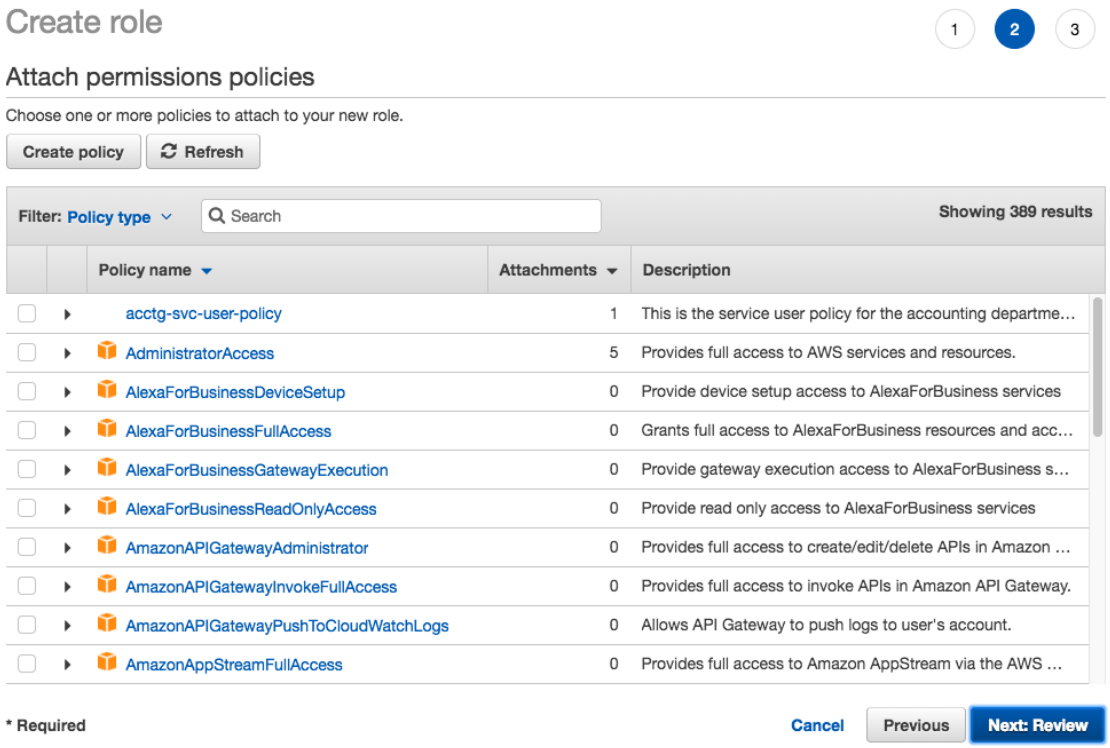

Figure 3.9 – The Attach permissions policies dialog.

When the `Attach permissions policies` dialog (shown in Figure 3.9) opens, do not select a policy; instead, click `Next:Tags`, then `Next: Review` to continue.

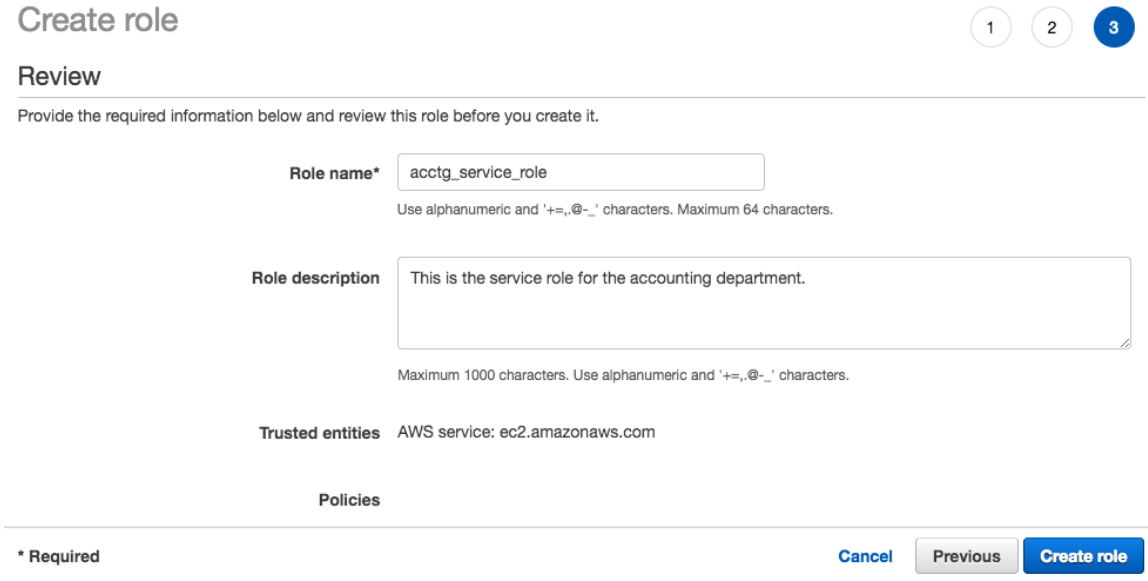

Figure 3.10 - Provide a role name and description.

When the `Review` dialog opens (shown in Figure 3.10), specify a name and description for the new role and click the `Create role` button. The new role will be displayed in the role list on the Amazon IAM `Roles` page. Click the role name to display detailed information about the role on the `Summary` dialog.

The `Summary` dialog will display a `Role ARN`, but the ARN will not be enabled until the security policy and trust policy are updated. To modify the inline security policy, click the `Add inline policy` button; the button is located on the `Permissions` tab (see Figure 3.11).

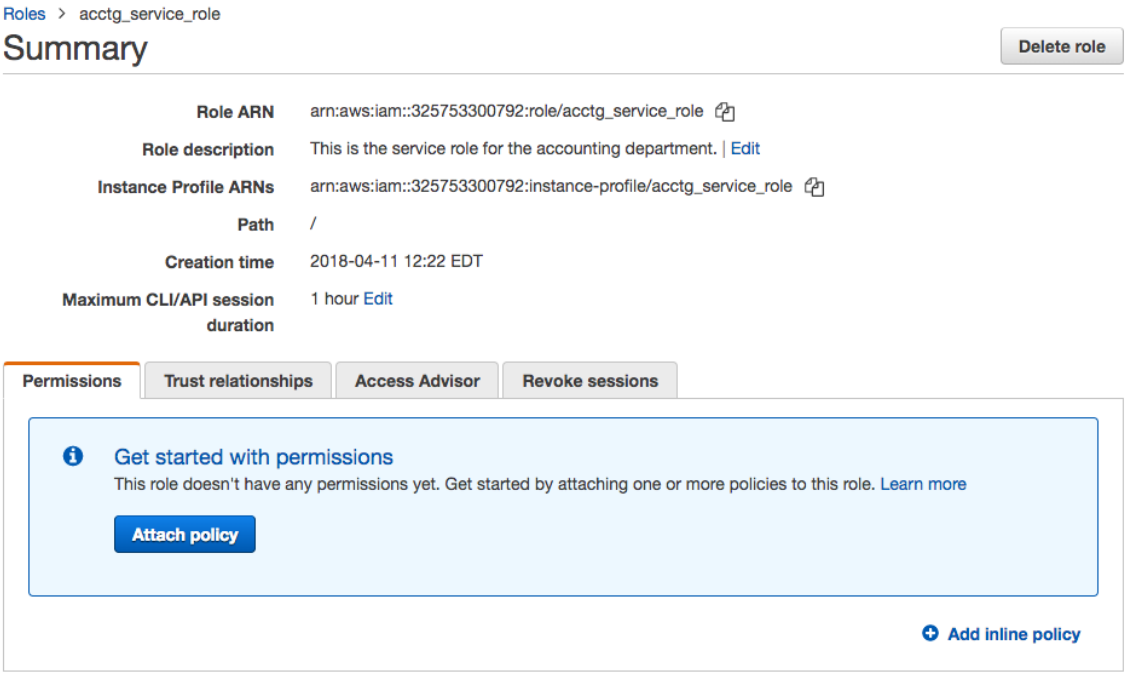

Figure 3.11 - Add a Custom Policy.

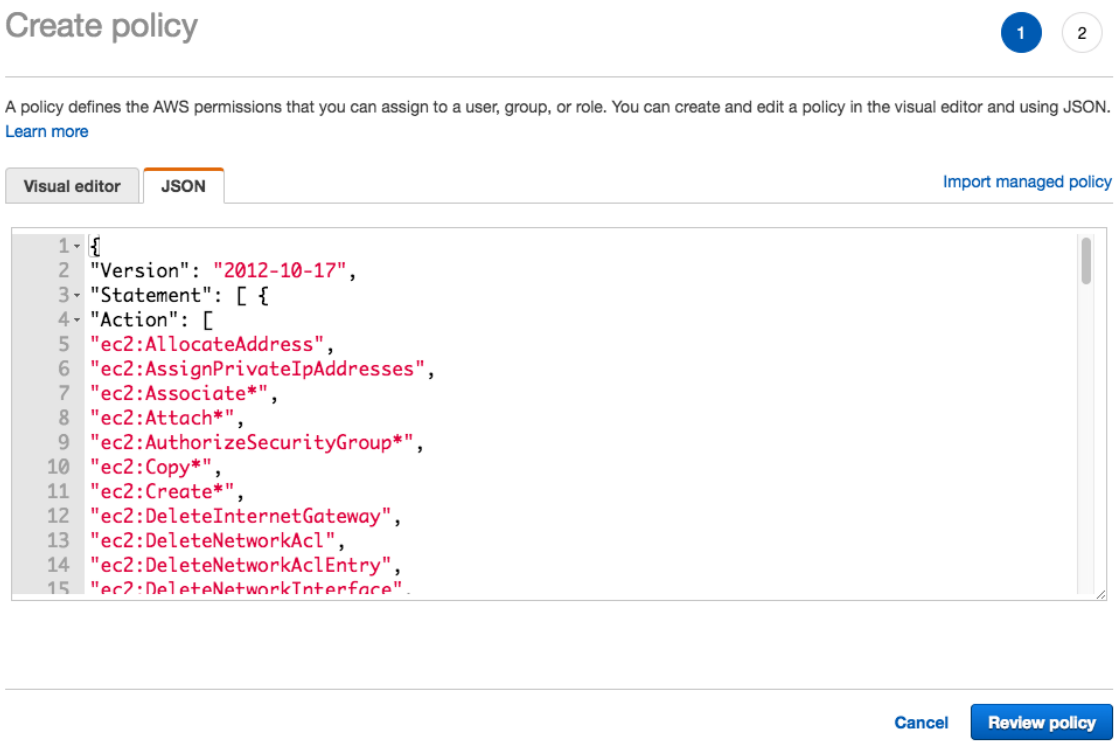

Figure 3.12 - Provide the policy name and contents.

Copy the security policy text into the the `JSON` tab on the `Create policy` dialog (see Figure 3.12). For a security policy that you can use when creating the service role, please see Section 10.3.

After providing security policy information, click `Review Policy` to provide a name for the policy, and return to the role information page.

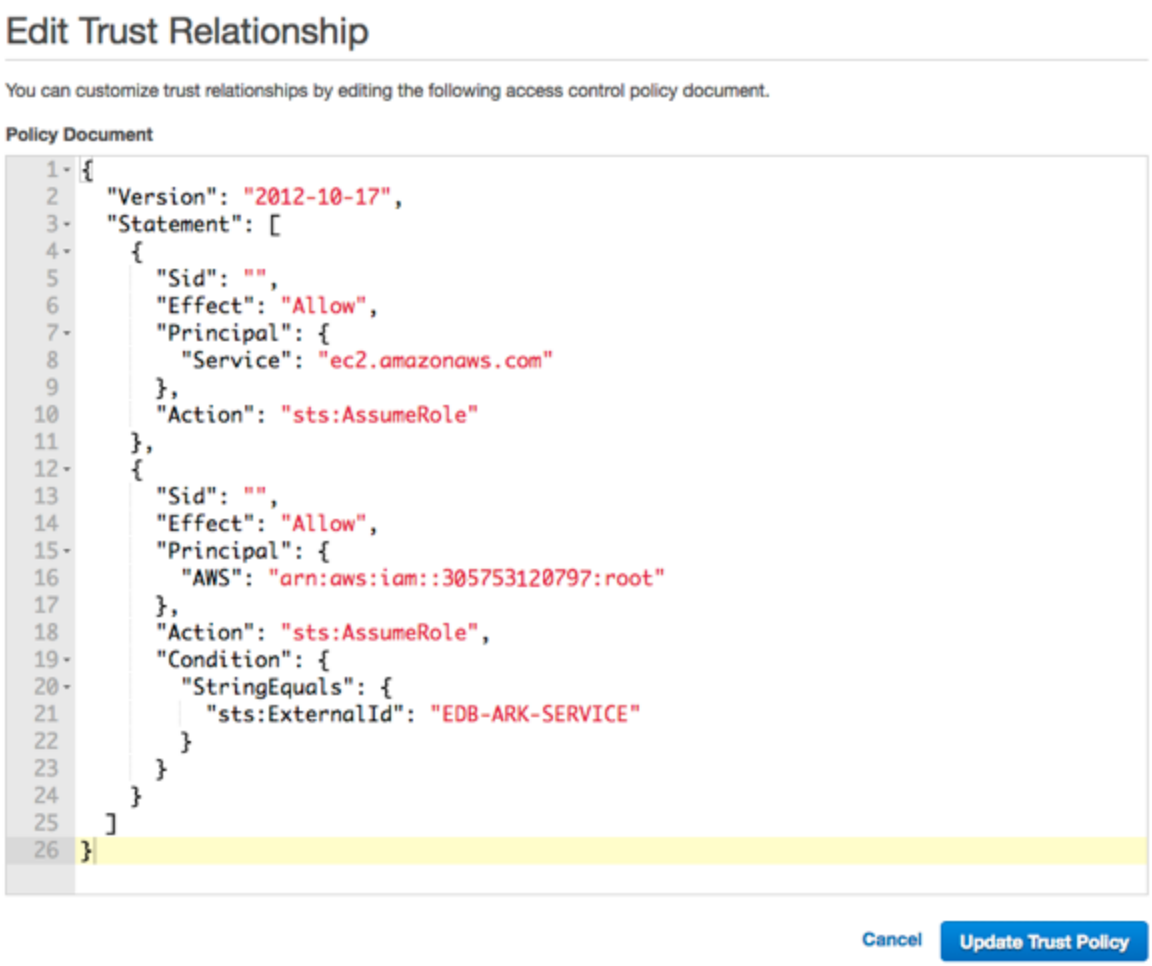

Figure 3.13 - The Policy Document.

Navigate to the `Trust relationships` tab, and select the `Edit Trust Relationship` button to display the `Policy Document` (see Figure 3.13). Replace the displayed content of the policy document with the content of the security policy included in 0.

Click the `Update Trust Policy` button to finish.

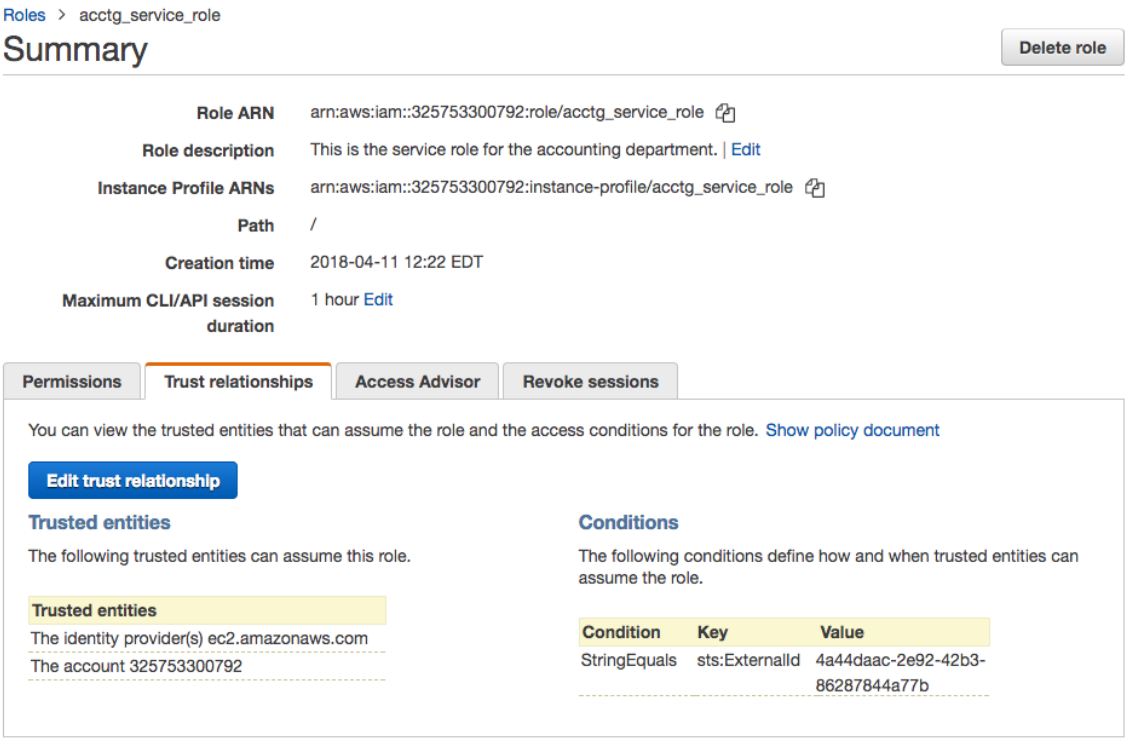

Figure 3.14 - The Summary tab of the Role detail panel.

The `Summary` dashboard (see Figure 3.14) 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. You can find this value under the `Conditions` section of the `Trust Relationships` tab.

3.1.3 Configuring the Ark Console

After launching the instance in the Amazon console, navigate to the public IP address of the cluster.

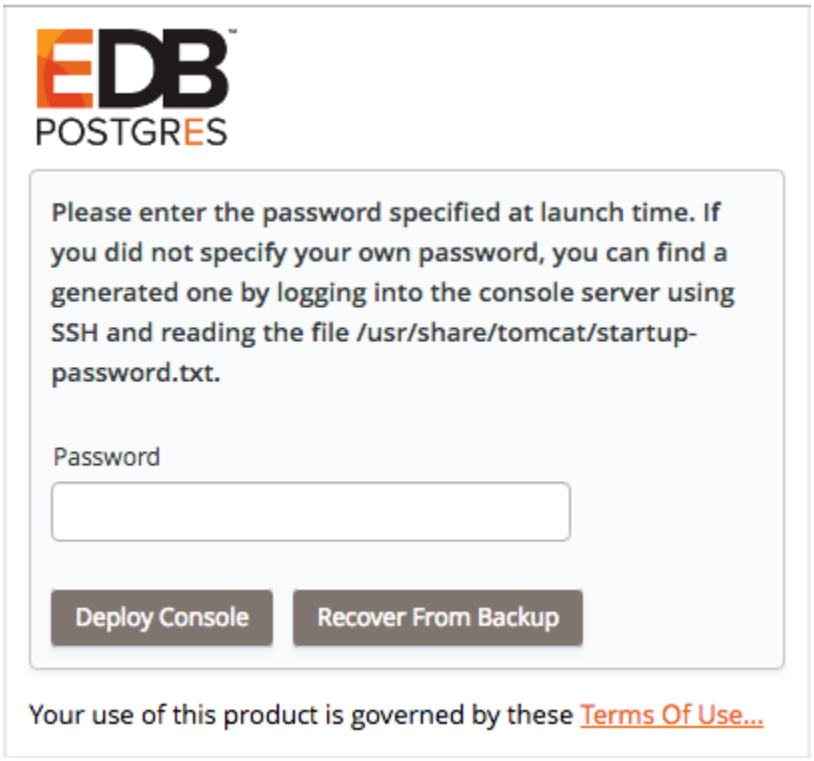

Figure 3.15 - Deploying the console.

The browser will open to the EDB Postgres `Password` dialog. When prompted, provide the password specified when launching the console (see Figure 3.15), and click `Deploy Console`.

Please note: You can use the `Admin` tab of the Ark console to access a dialog that will allow you to modify many of the deployment properties; for details, see Section 4.1.10.

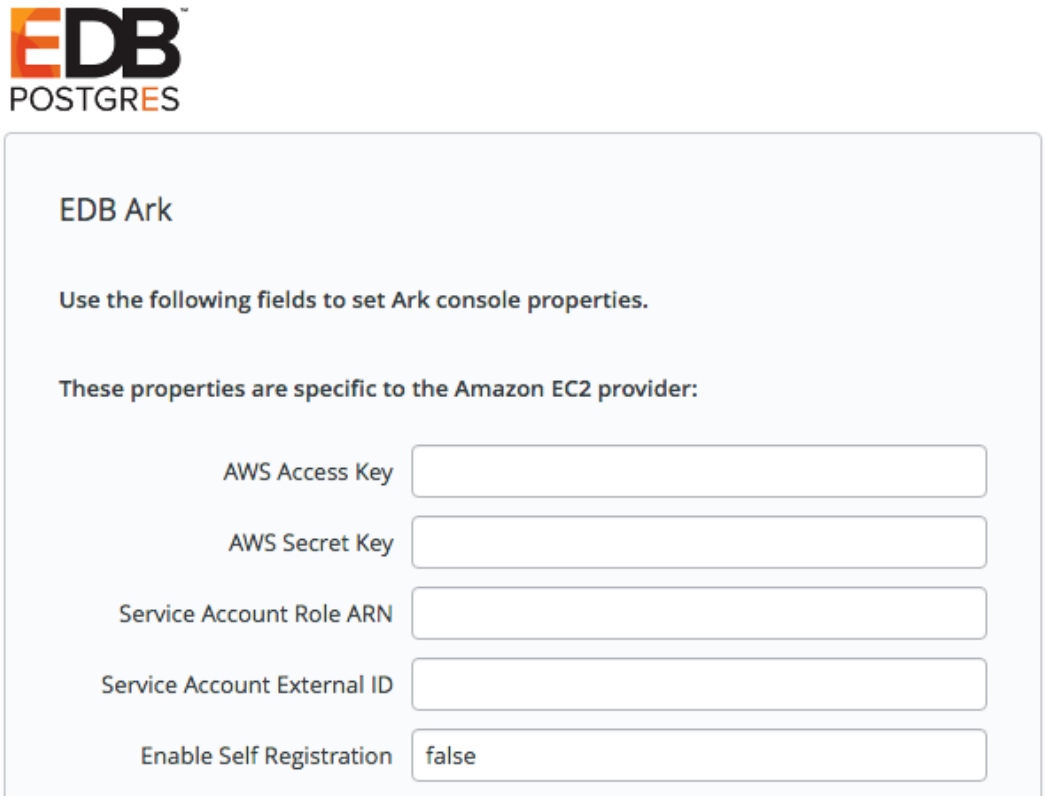

Figure 3.16 – The platform specific console properties.

Use fields in the first section of the dialog to provide values that are specific to your Amazon account:

- 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 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.
- 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.

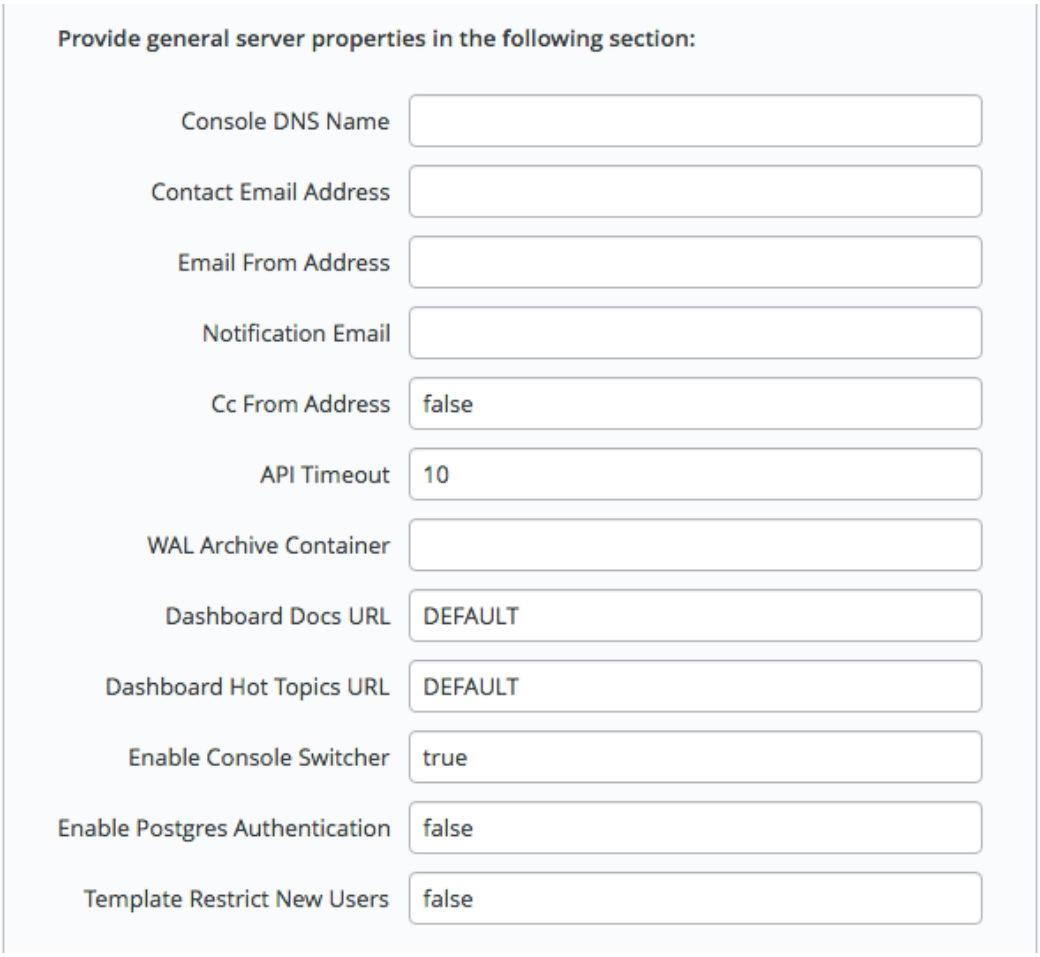

Figure 3.17 – The general server property fields.

Use fields in the next section to provide general server properties:

- Use the `Console DNS Name` field to specify a custom DNS name for the console. The property does not assign the DNS name to the console, but any notification emails that refer to the console will refer to the console by the specified name. If you do not provide a custom DNS name, the IP address of the console will be used in notifications.

- 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.

- Set the `CC From Address` field to `true` to instruct Ark to send a copy of the email to the `Email From Address` anytime a notification email is 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 `Template Restrict New Users` field to configure the Ark console to make any new user a Template Only user by default. You can later modify the user definition in the `User Administration` table to specify that a user is not a template only user.

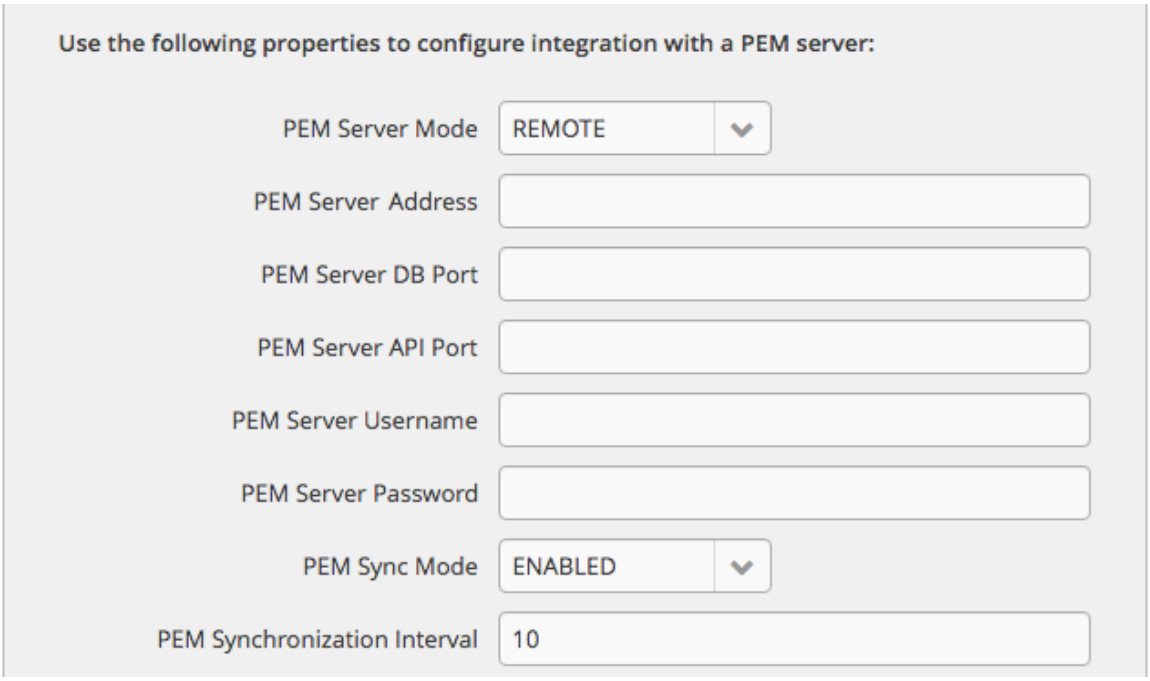

Figure 3.18 – The PEM server fields.

Use fields in the next section to provide connection details for a PEM server host; this will allow Ark to register and unregister PEM agents and clusters:

- Use the PEM Server Mode drop-down listbox to select a deployment mode:

  Select `DISABLE` to indicate that clusters deployed on the host should not be registered with the PEM server.

  Select `LOCAL` to indicate that you would like to use the PEM server that resides on your local host. If you select `LOCAL`, the PEM deployment will use default values assigned by the installer.

  - The IP address of the PEM server host will be the IP address of the Ark host.
  - The `PEM Server Port` will monitor port `5432`.
  - The PEM server database user will be named `postgres`.

- The password associated with the PEM server will be the same password as the Ark console.

Select `REMOTE` to indicate that you would like to use a PEM server that resides on another host, and provide connection information on the Ark console deployment dialog. If you select `REMOTE`, whenever a new cluster node is created on this console, it will be registered for monitoring by the PEM server.

- Provide the host name or IP address of the PEM server host in the `PEM Server Address` field.
- Specify the port monitored for connections by the PEM server in the `PEM Server DB Port` field.
- Specify the port on the PEM server host used for PEM API connection attempts by the Ark server in the `PEM Server API Port` field. Not valid if the PEM server mode is `DISABLED` or `LOCAL`.
- Provide the name that should be used when authenticating with the PEM server in the `PEM Server Username` field.
- Provide the password associated with the PEM server user in the `PEM Server Password` field.

- Use the `PEM Sync Mode` drop-down listbox to `ENABLE` or `DISABLE` synchronization between the Ark server and the PEM server. For more information about syncing with the PEM server, see Section 2.2.1.
- Use the `PEM Synchronization Interval` field to specify the number of minutes between attempts to synchronize the Ark console with the PEM server.

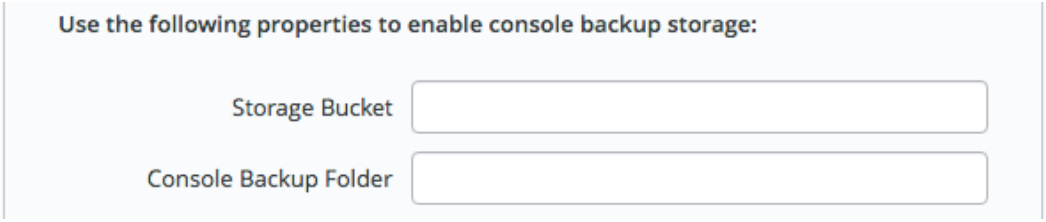

Figure 3.19 – The console backup storage fields.

Use fields in 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.

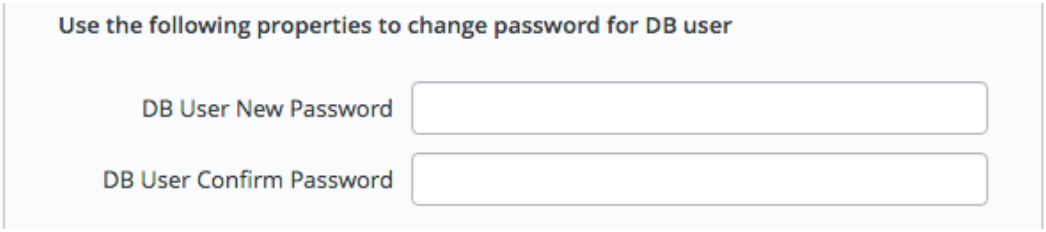

Figure 3.20 – The database password fields.

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`).

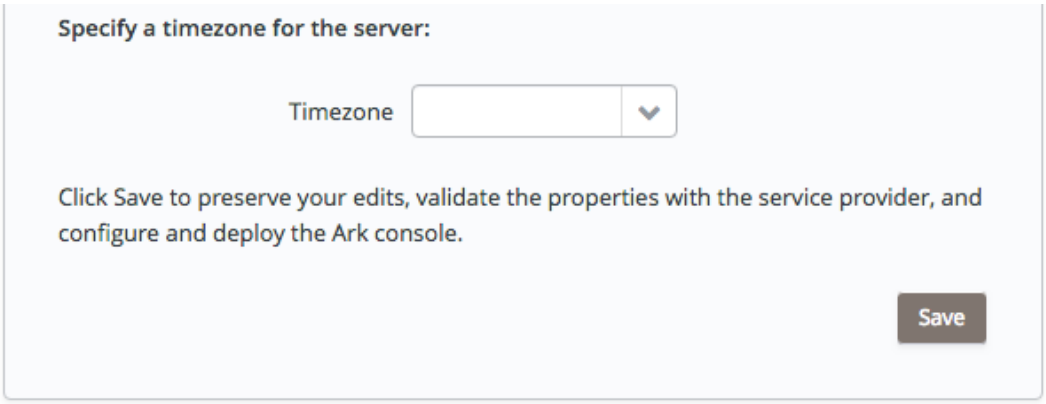

Figure 3.21 – The timezone field.

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. The Ark console will prompt you to confirm that you wish to restart the server; when prompted, click the `Restart` button to restart the server and start the Ark 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 through the `Identity and Access Management` dashboard to the `Roles` dashboard, and click the `Create role` button.

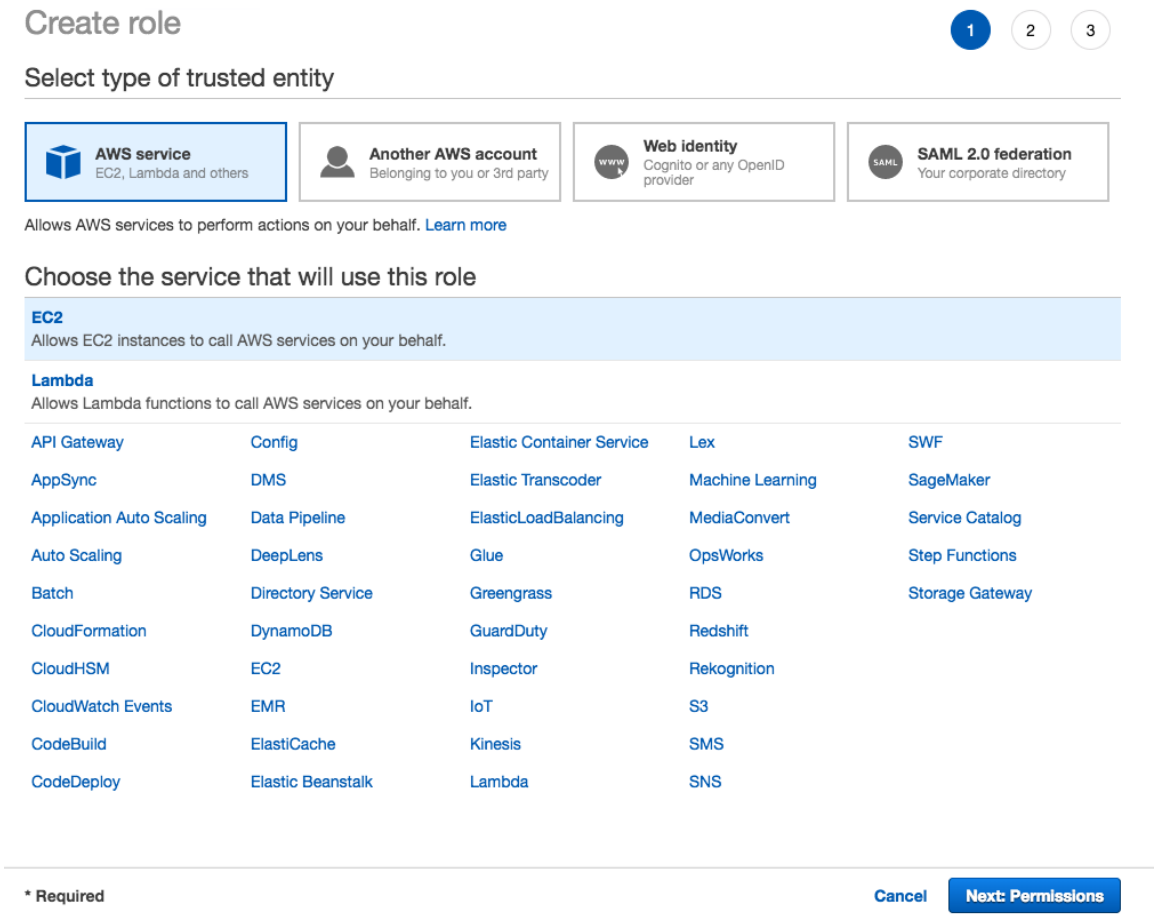

Figure 3.22 - The Create role dialog.

When the `Create role` dialog opens (shown in Figure 3.22), select the AWS service button and highlight the EC2 bar, and click `Next: Permissions` to continue.

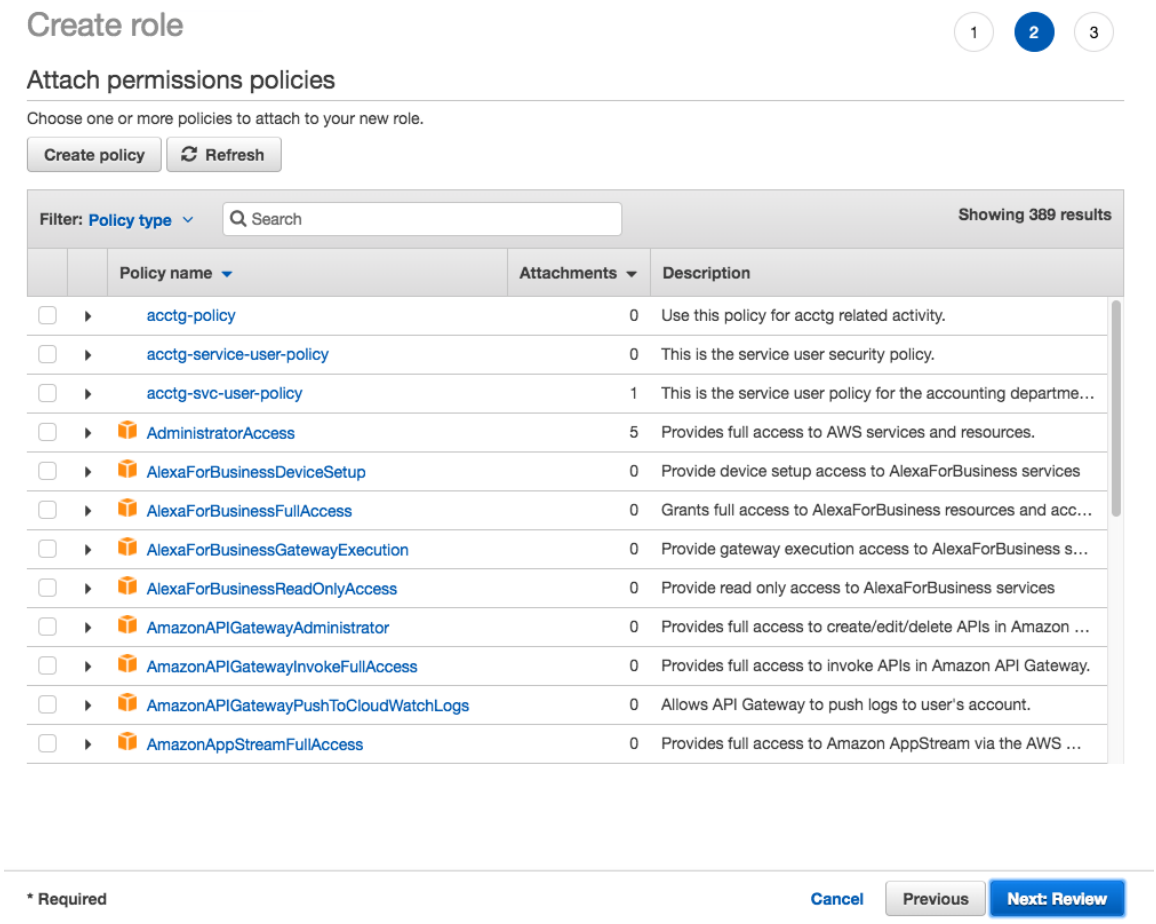

Figure 3.23 – The Attach permissions policies dialog.

When the `Attach permissions policies` dialog opens (see Figure 3.23), do not specify a policy; instead, click `Next: Review` to continue.

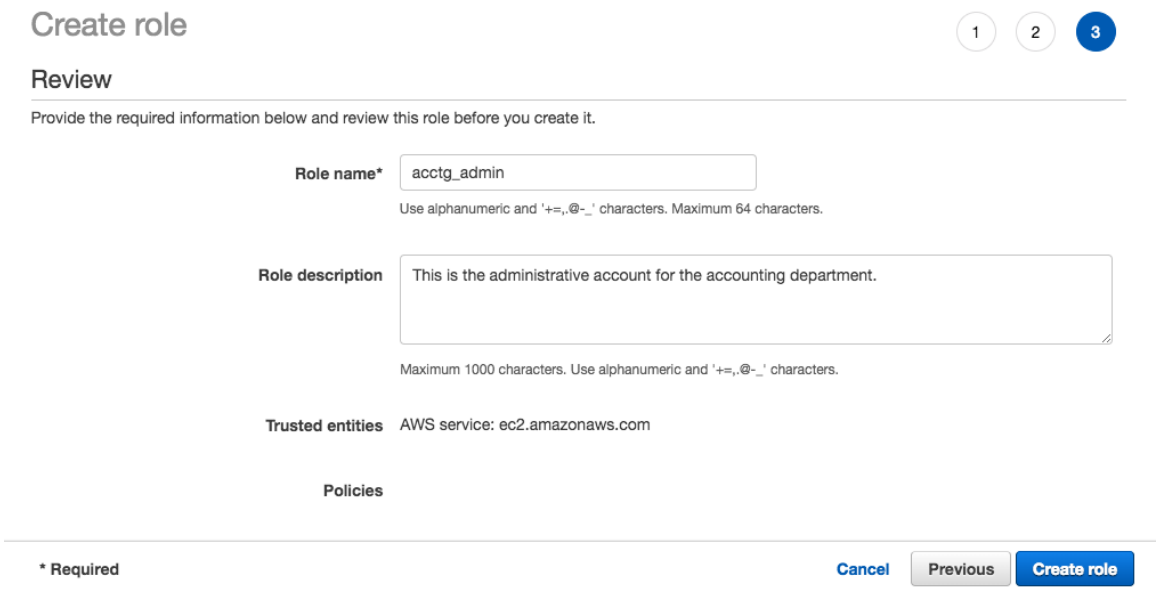

Figure 3.24 – The Review dialog.

Use the `Review` dialog (see Figure 3.24) to provide a name and a description; then, click `Create role`. The role will be displayed in the role list on the Amazon IAM `Roles` page. Highlight the role name to review account details (see Figure 3.25).

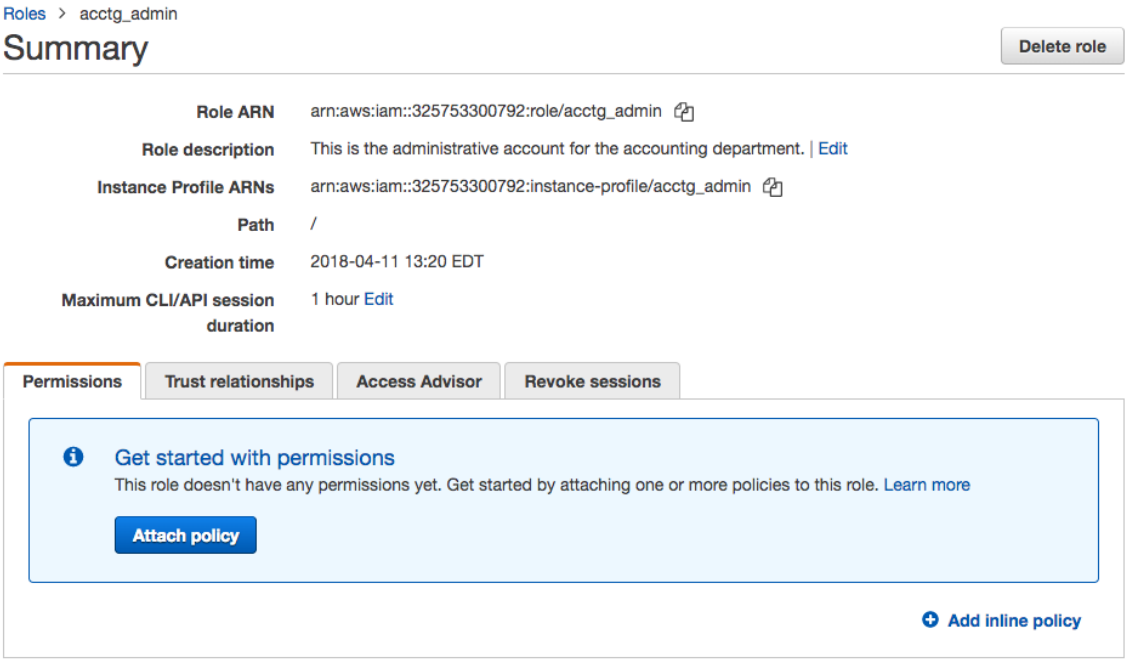

Figure 3.25 – The Summary dialog.

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. Click the Add inline policy button to add a security policy.

Copy the permission policy text into the JSON tab (see Figure 3.26). The permission policy required by Ark is available in Section 10.3.

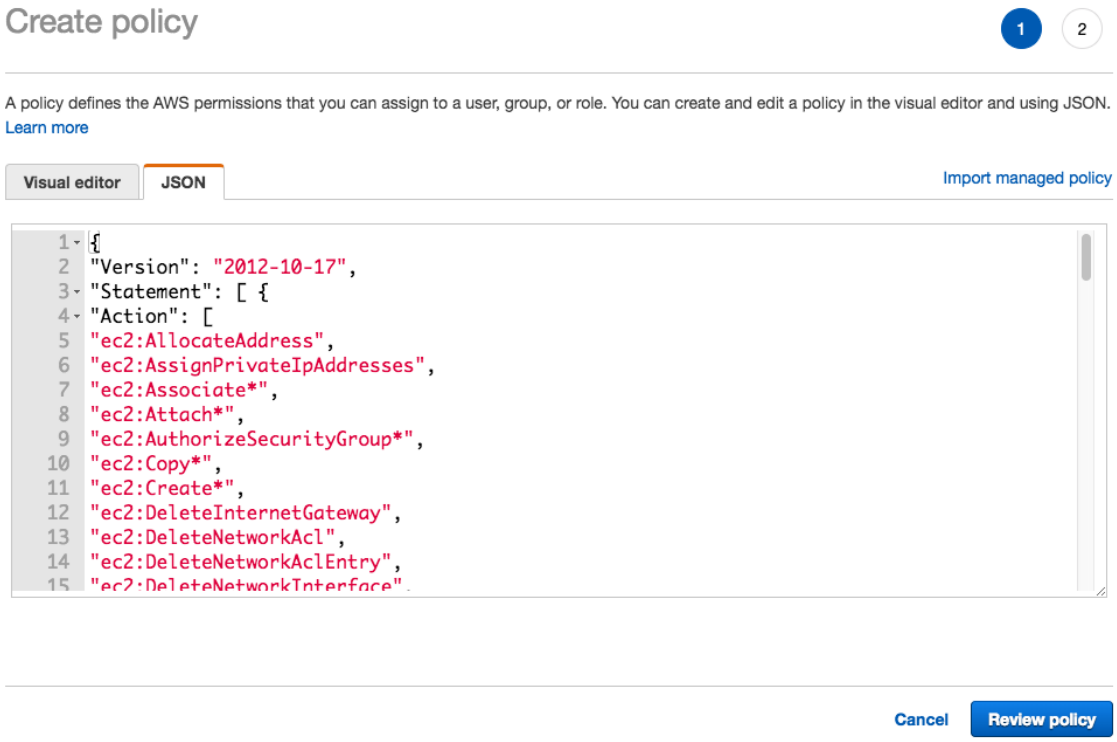

Figure 3.26 – Adding a security policy.

Then, click Review Policy to return to continue to the Review policy page and provide a name for the policy. Then, click the Create policy button to return to the role summary page.

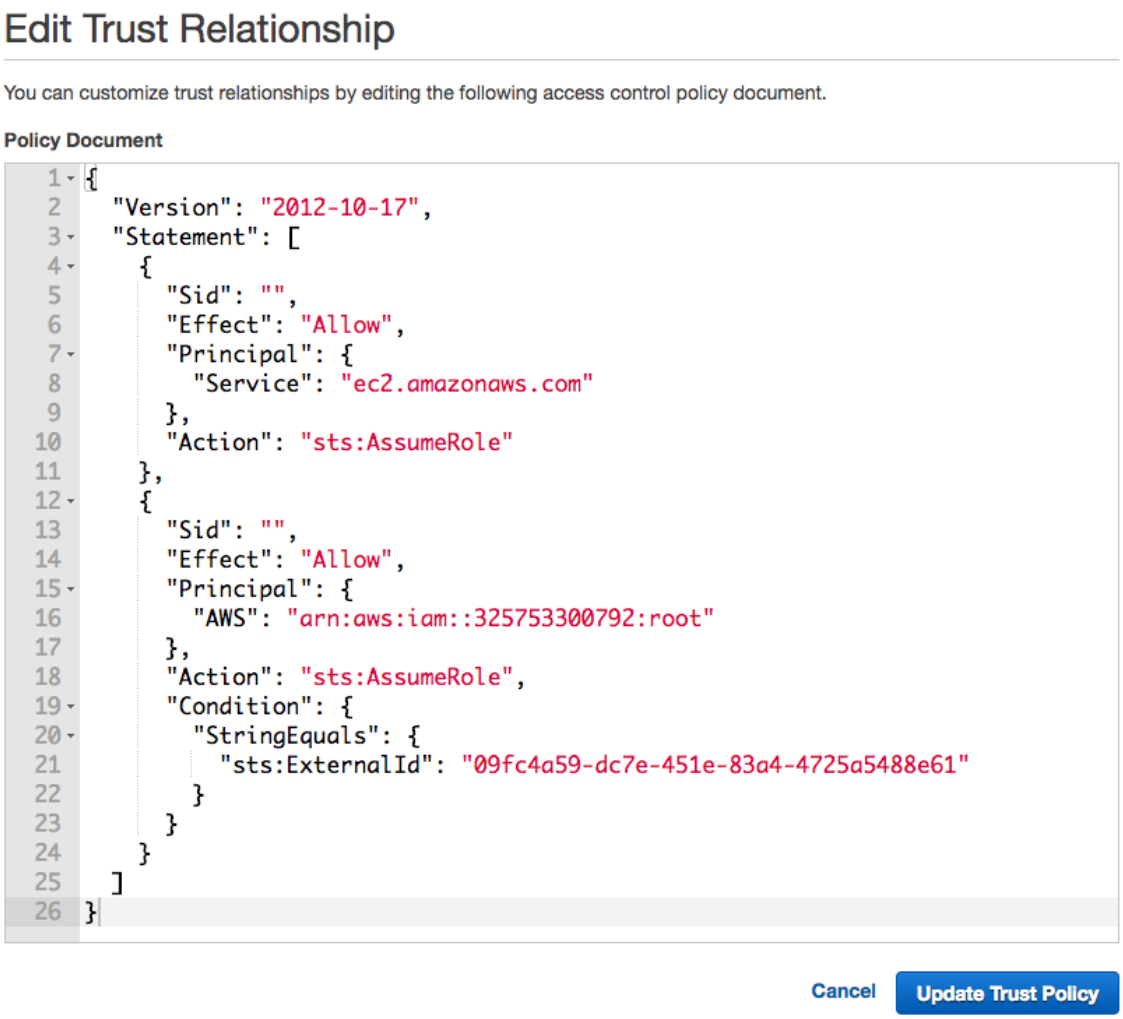

Figure 3.27 - Editing the trust relationship.

Select the `Trust relationships` tab, and click the `Edit trust relationship` button to update the trust relationship assigned to the role (see Figure 3.27). Replace the displayed content of the policy document with the content of the file available in Section 10.2.

Please note: `EDB-ARK-SERVICE` is a placeholder within the trust policy. 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.

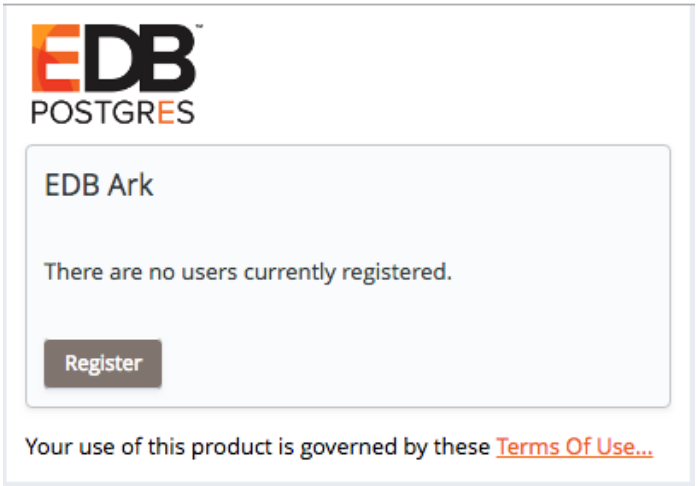

Figure 3.28 - Accessing the New User Registration dialog.

Click the Register button to open the New User Registration dialog (shown in Figure 3.28).

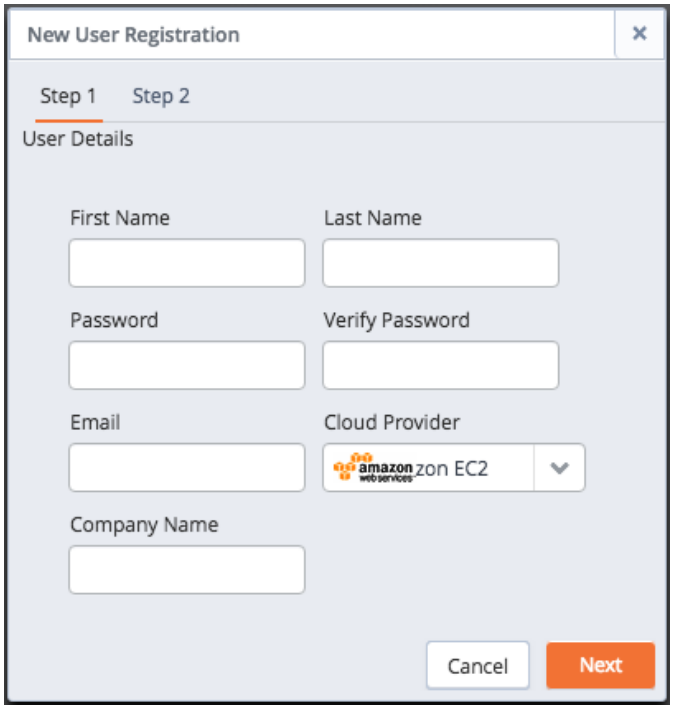

Figure 3.29 - The New User Registration dialog.

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.

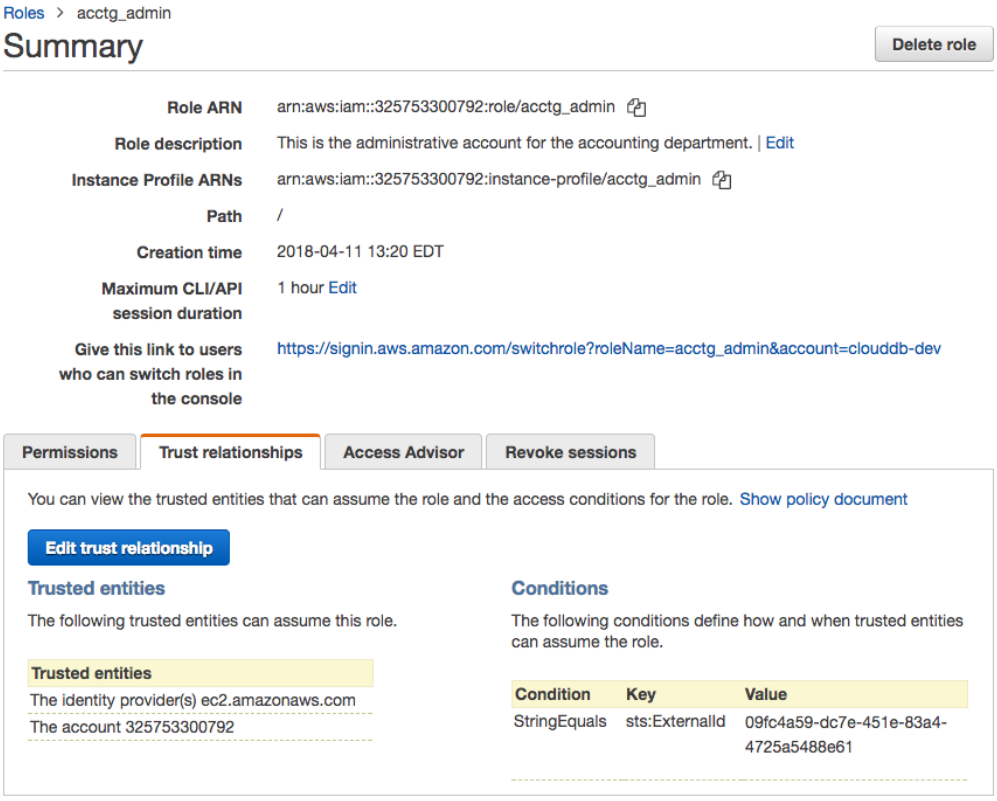

Figure 3.29 - The Summary tab of the Role detail panel.

Your Amazon IAM role ARN is displayed on the Amazon role detail panel (see Figure 3.29).

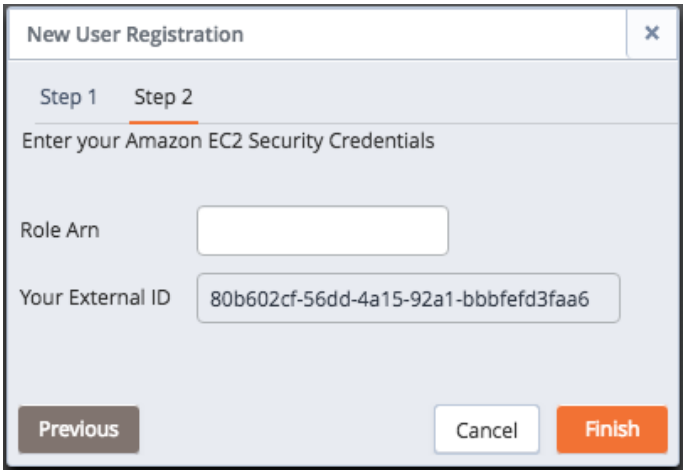

Figure 3.30 - Registering a user on an Amazon EC2 cloud.

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 3.30). 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 3.31).

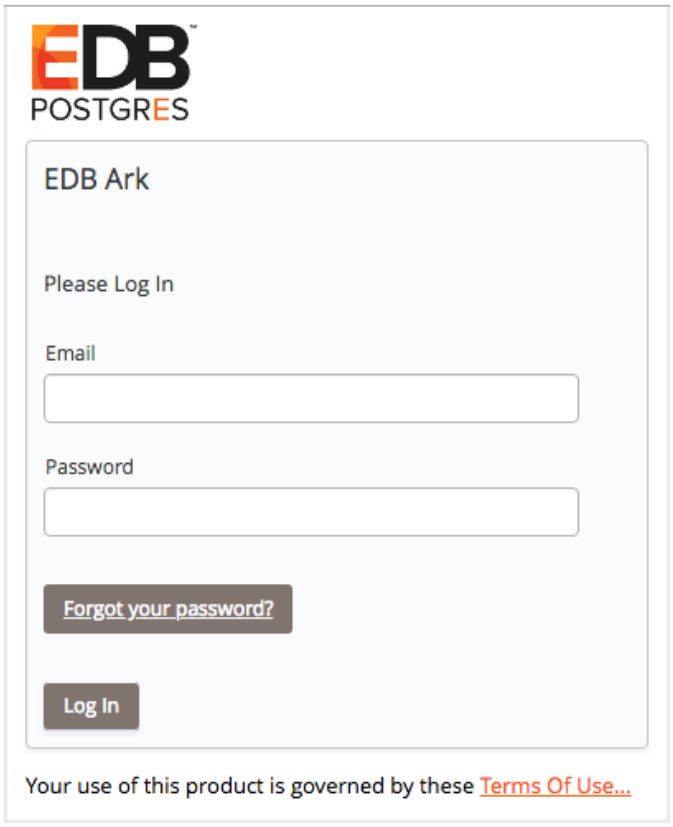

Figure 3.31 - The Login/Register dialog.

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 3.32).

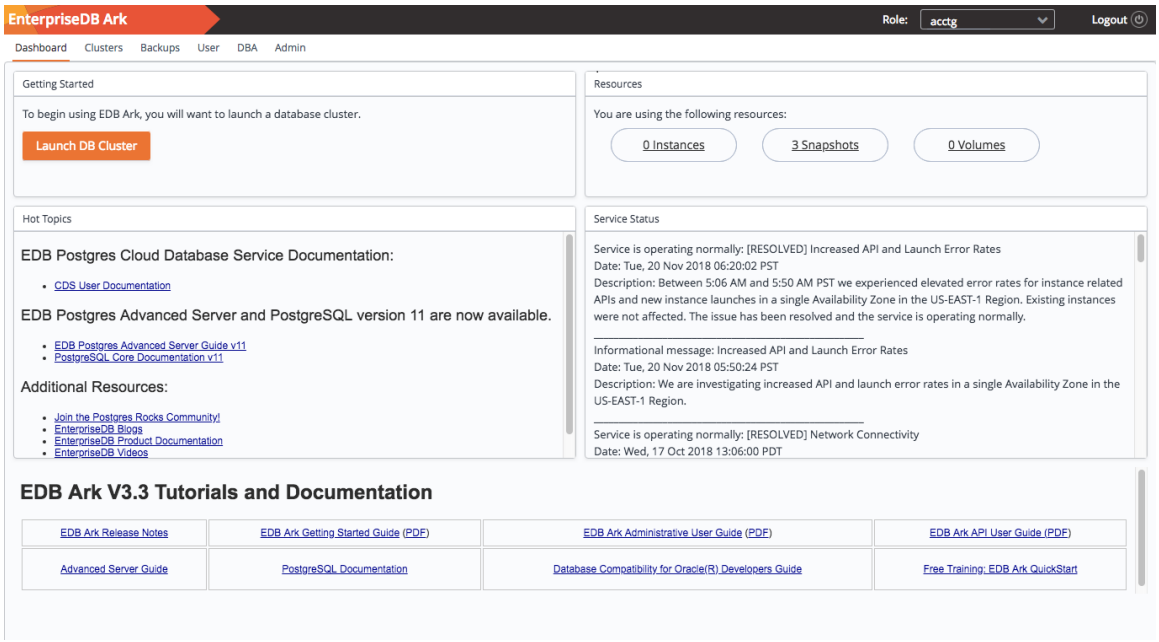

Figure 3.32 - The Dashboard tab of the Ark management console.

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 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 3.2.1.
- Create an Azure network security group. For more information, see Section 3.2.2.
- Create an Azure storage account. For more information, see Section 3.2.3.
- Launch a VM Image that contains the Ark console. For more information, see Section 3.2.4.
- Configure the Ark console. For more information, see Section 3.2.5.
- Register an Ark Administrative user. For more information, see Section 3.2.6.

3.2.1 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

3.2.2 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
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
Custom TCP	Ingress	5432	CIDR	0.0.0.0/0

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.

3.2.3 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.

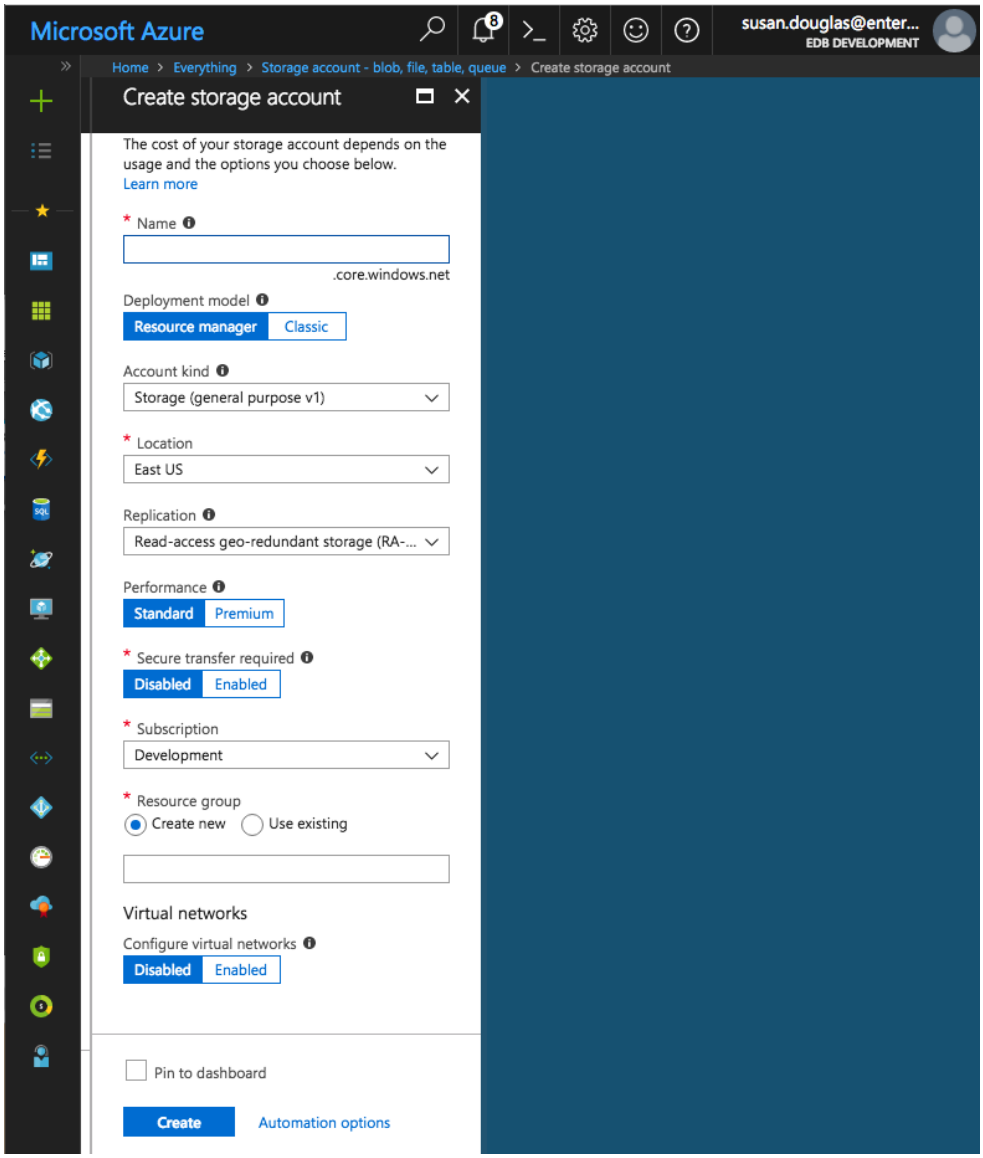

Figure 3.52 – Defining a storage account.

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 3.52).

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/

3.2.4 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.

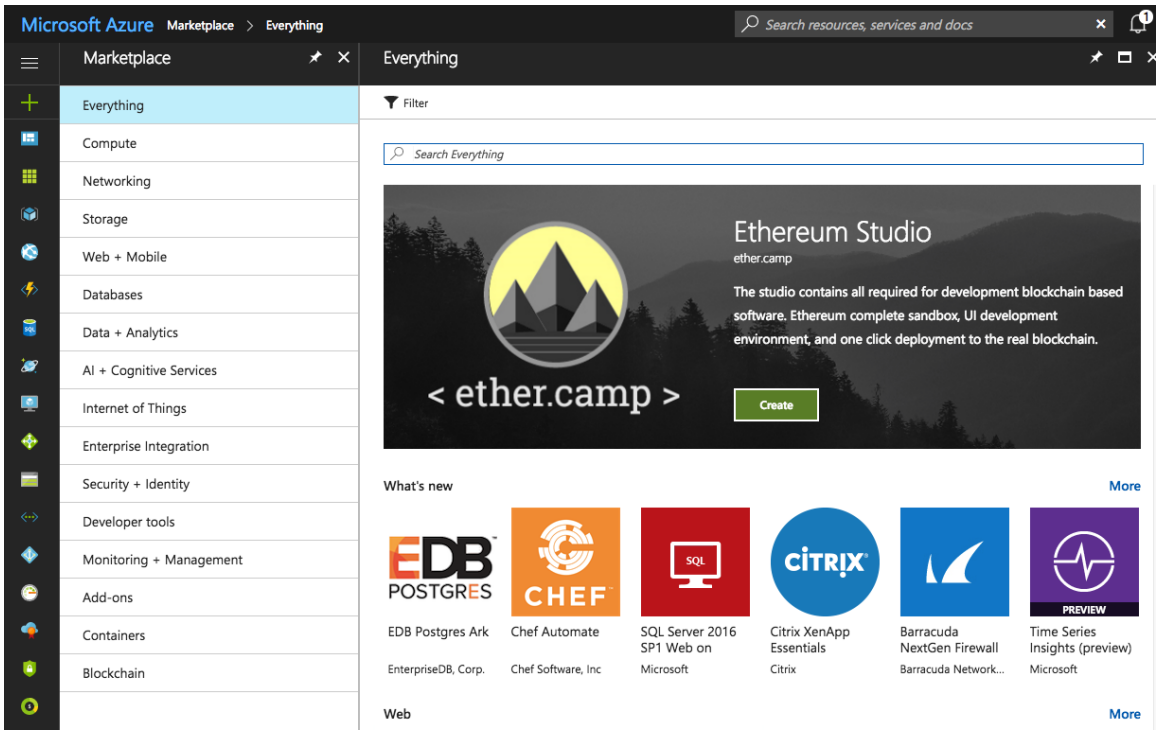

Figure 3.53 – Selecting an image.

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.

The Azure console will open to a dialog that allows you to configure the virtual machine that will host your console deployment. Please note that your virtual machine requirements may vary from the description that follows; the description is intended to provide guidelines about the minimum requirements for a console host for an Ark deployment. Please consult the Azure documentation for detailed information about additional configuration options for your virtual environment.

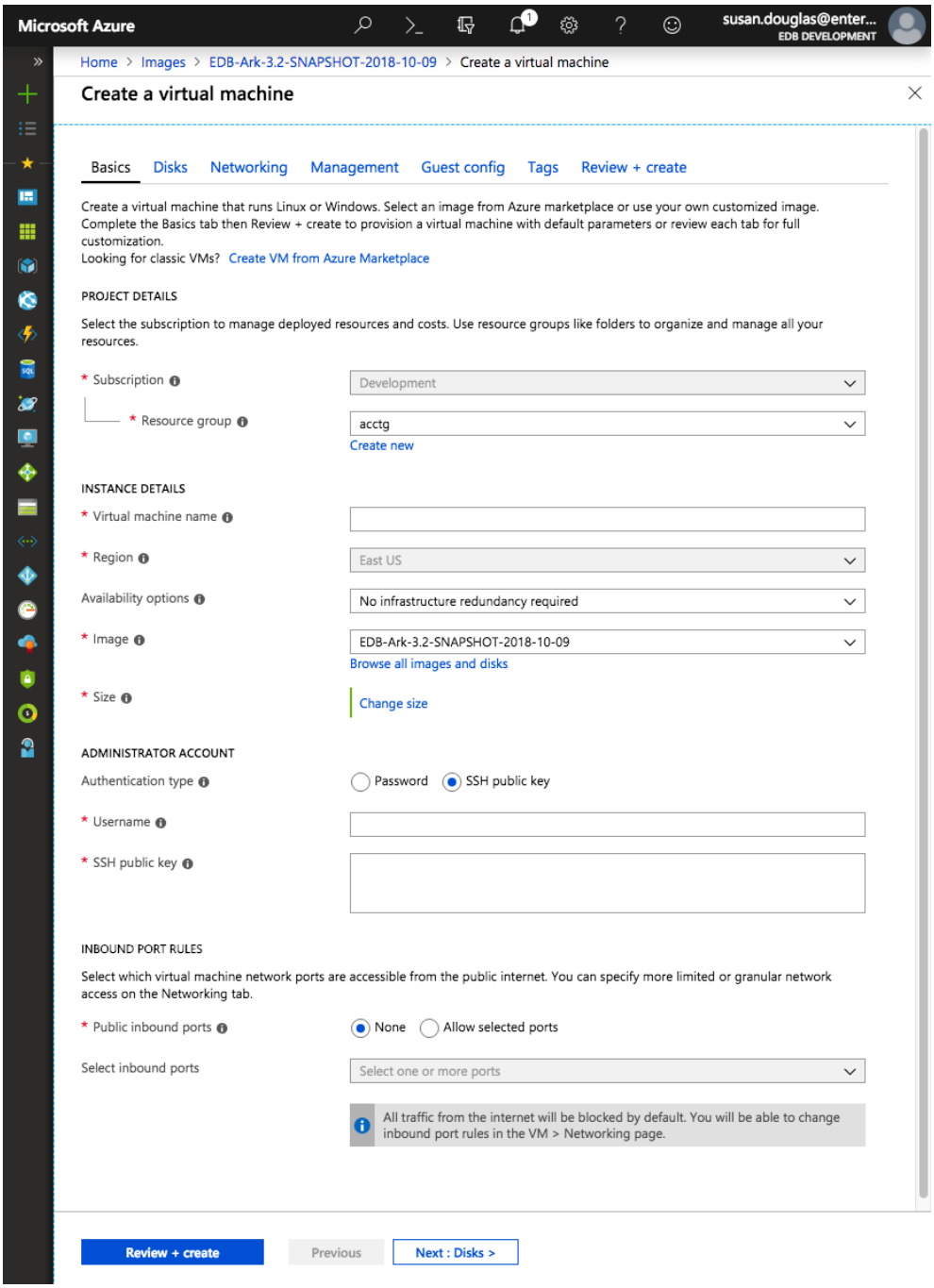

Figure 3.54 – Creating a virtual machine.

Use fields on the `Basics` panel (see Figure 3.54) to provide general information about the new virtual machine:

- If applicable, use the `Subscription` drop-down listbox to select the name of an Azure subscription.

- Use the `Resource group` drop-down listbox to select the resource group in which the VM will be created.

- Provide a name for the VM in the `Virtual Machine Name` field.

- If applicable, use the `Region` drop-down listbox to select the region in which the VM will reside.

- Use the `Image` drop-down listbox to select the image that will be used for the VM.

- Use the `Change size` link (in the `Size` field) to open the `Select a VM size` panel (see Figure 3.55) and select the machine configuration.

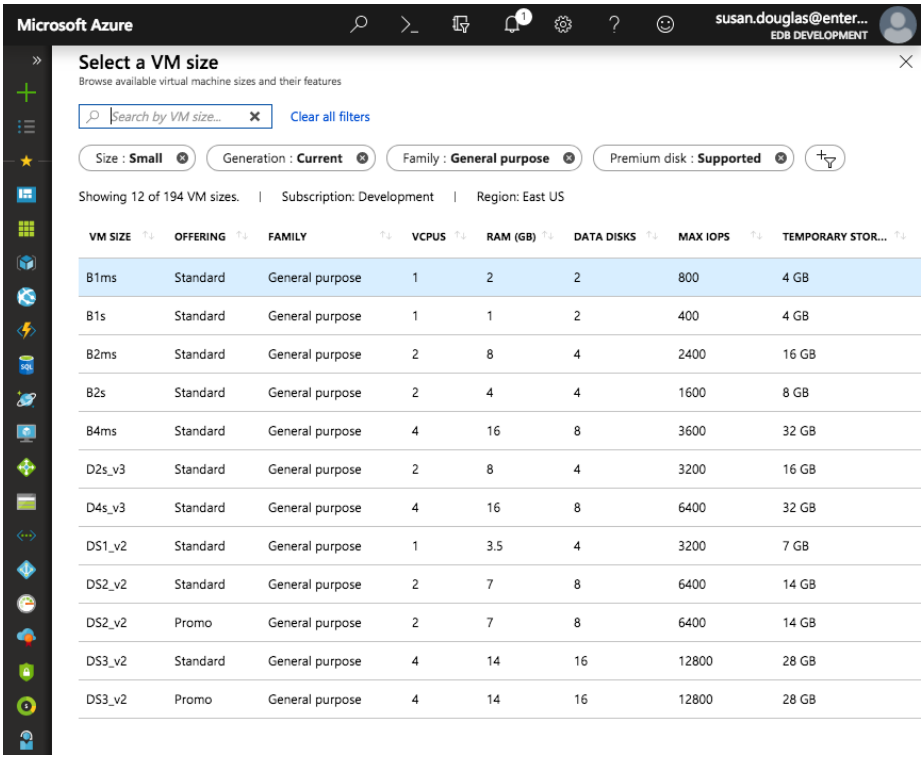

Figure 3.55 – Selecting a machine size.

Highlight a configuration type, and click the `Select` button (in the lower-left corner of the panel) to continue.

- Select the radio button next to the `Authentication type` you wish to use for the Administrator account; we highly recommend using `SSH authentication`.

- Provide an operating system user name in the `User name` field; the default operating system user name for Ark images is `centos`.

- If you have elected to enable SSH public key authentication, provide the key in the `SSH public key` field.

Use fields on the `Networking` panel to specify your network configuration preferences. When configuring an Azure virtual machine to use the Ark console, you should:

- Select the `Advanced` radio button in the `Network security group` field.

- Use the `Network security group` drop-down listbox to select the security group that you wish to use for the virtual machine.

Use fields on the `Guest config` panel to provide an extension that sets the Ark console deployment password. Create a file on your local system named `startup-password.sh` that contains the following text:

```
#!/bin/sh
rm -f /usr/share/tomcat/startup-password.txt
echo "console_password" > /usr/share/tomcat/startup-
password.txt
chown tomcat:tomcat /usr/share/tomcat/startup-password.txt
chmod 600 /usr/share/tomcat/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, click the `Select an Extension to install` link, then `Custom Script for Linux`. Then, click the `Create` button; use the fields on the `Install extention` panel (see Figure 3.56) to identify the script:

- Use `Script files` browser to locate and upload the script file.

- Enter the command that will invoke your script in the `Command` field; for example, `sh startup-password.sh`.

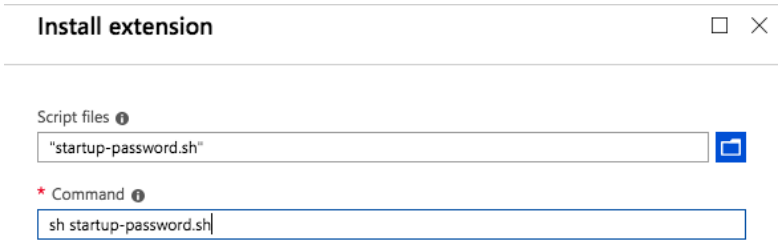

Figure 3.56 – Installing an extension.

Click `OK` to continue and return to the `Settings` panel; when you've finished updating the settings with your preferences, click `OK` to continue. Then, click the `Review + Create` button to validate your virtual machine definition (Figure 3.57).

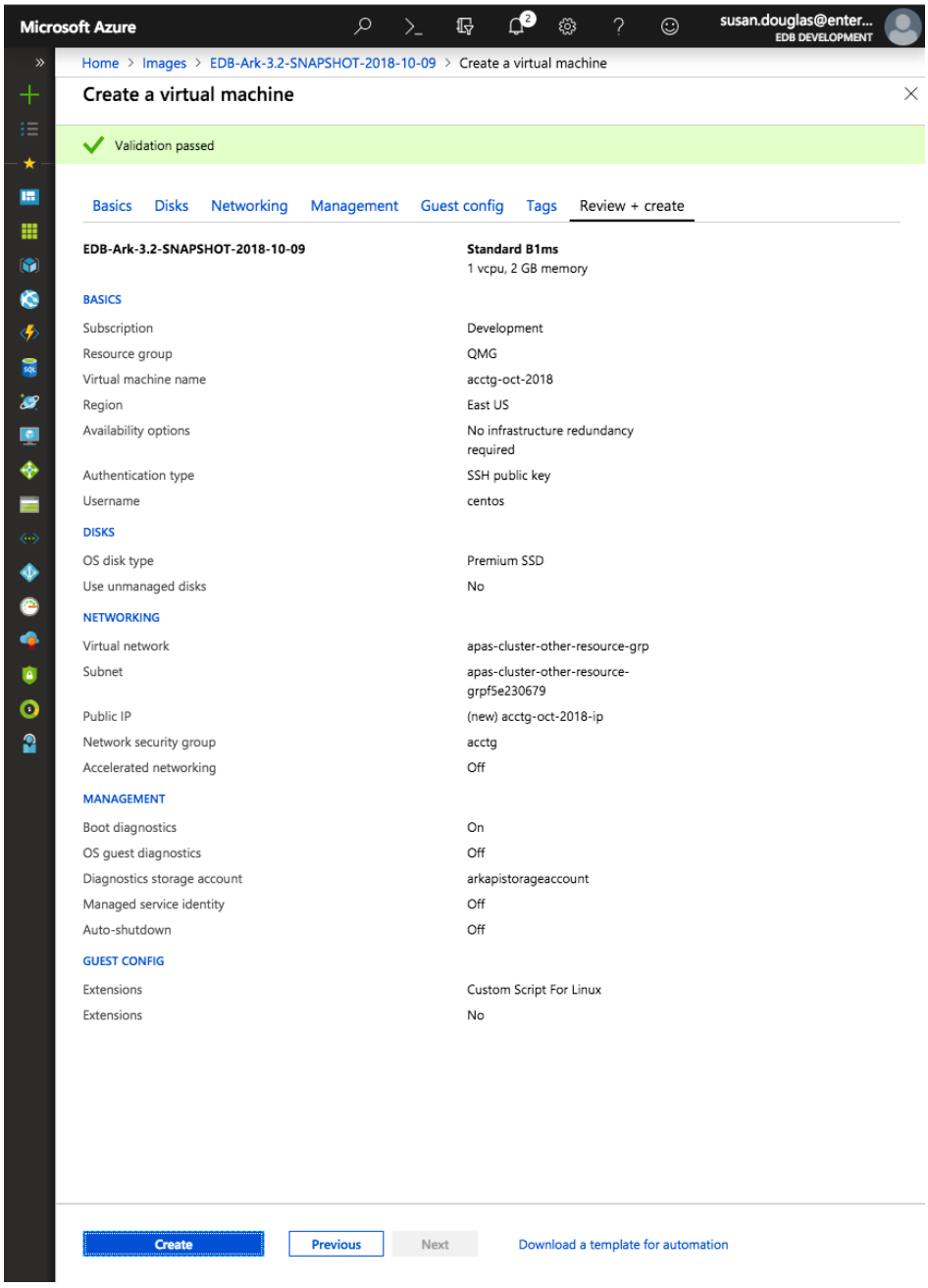

Figure 3.57 – Installing an extension.

Azure will confirm that your machine definition is valid; then, you can click the `Create` button to create your virtual machine.

You can monitor the virtual machine's deployment from the Azure `Operations` page, the `Resource group` activity log, or the `Virtual machine Overview` page.

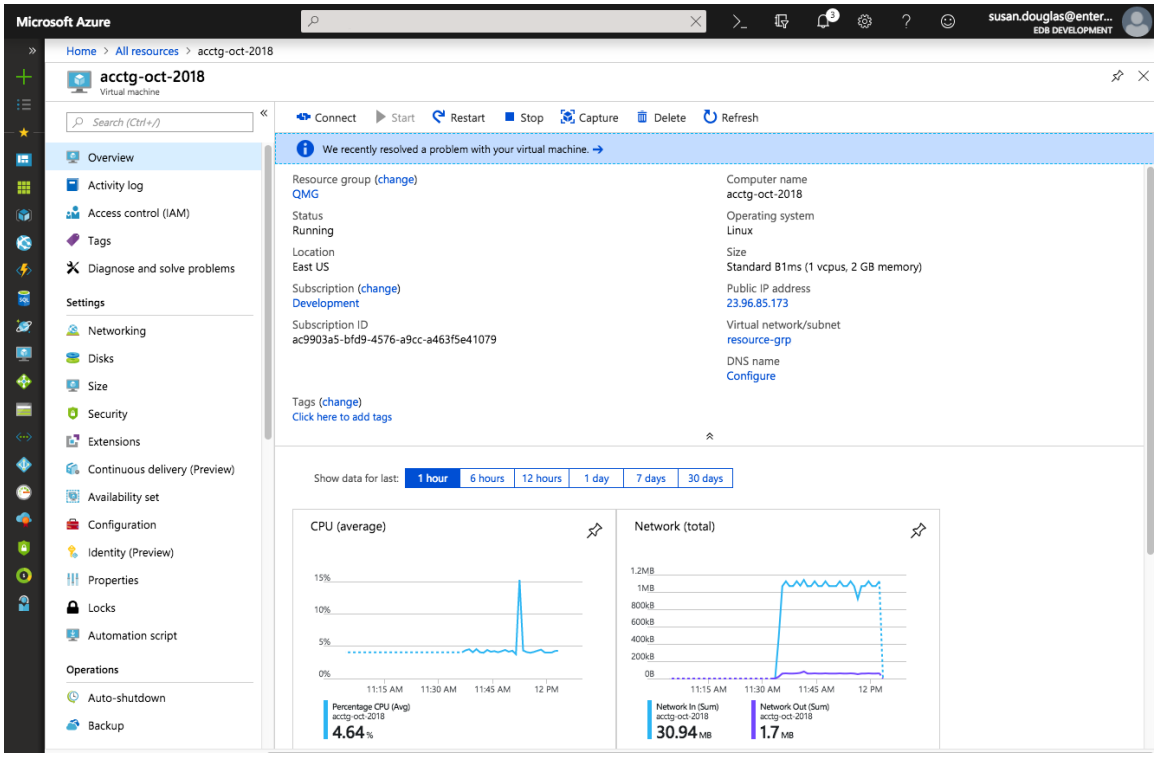

Figure 3.58 – The Virtual Machine details page.

While the deployment finishes, 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 details panel (see Figure 3.58).

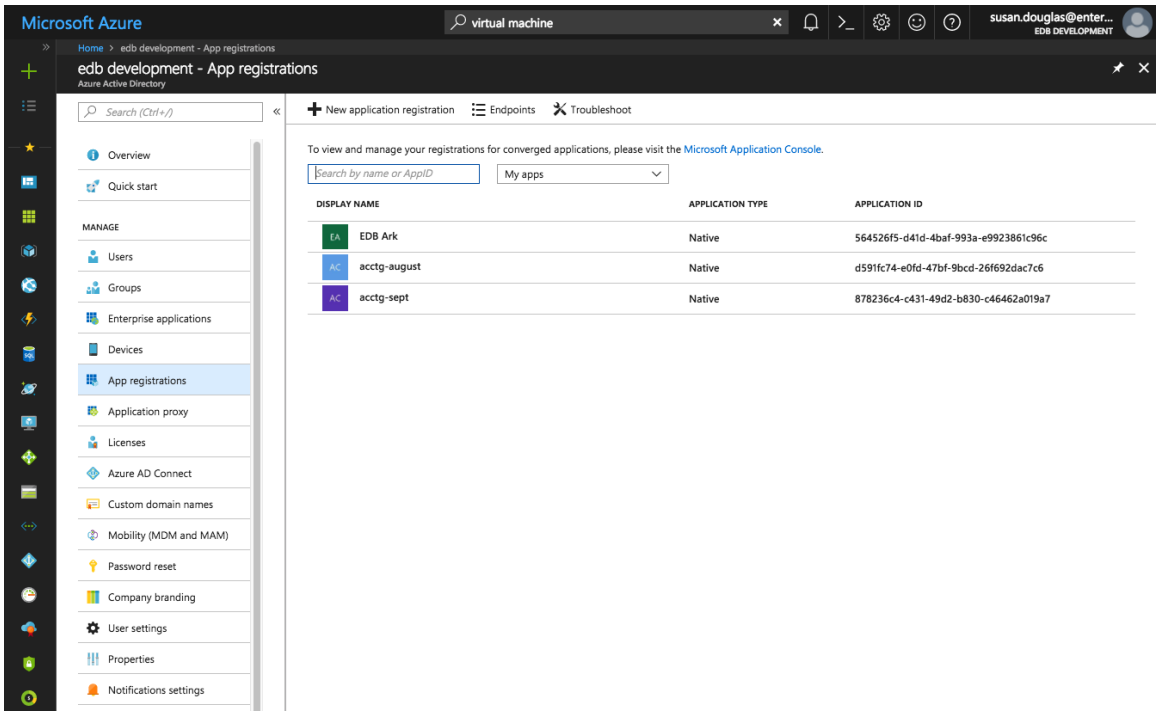

Figure 3.59 – The New application registration page.

After copying the public IP address of your server, select `App registrations` from the `Azure Active Directory` page. Click the `New application registration` button located on the `App registrations` detail panel (see Figure 3.59).

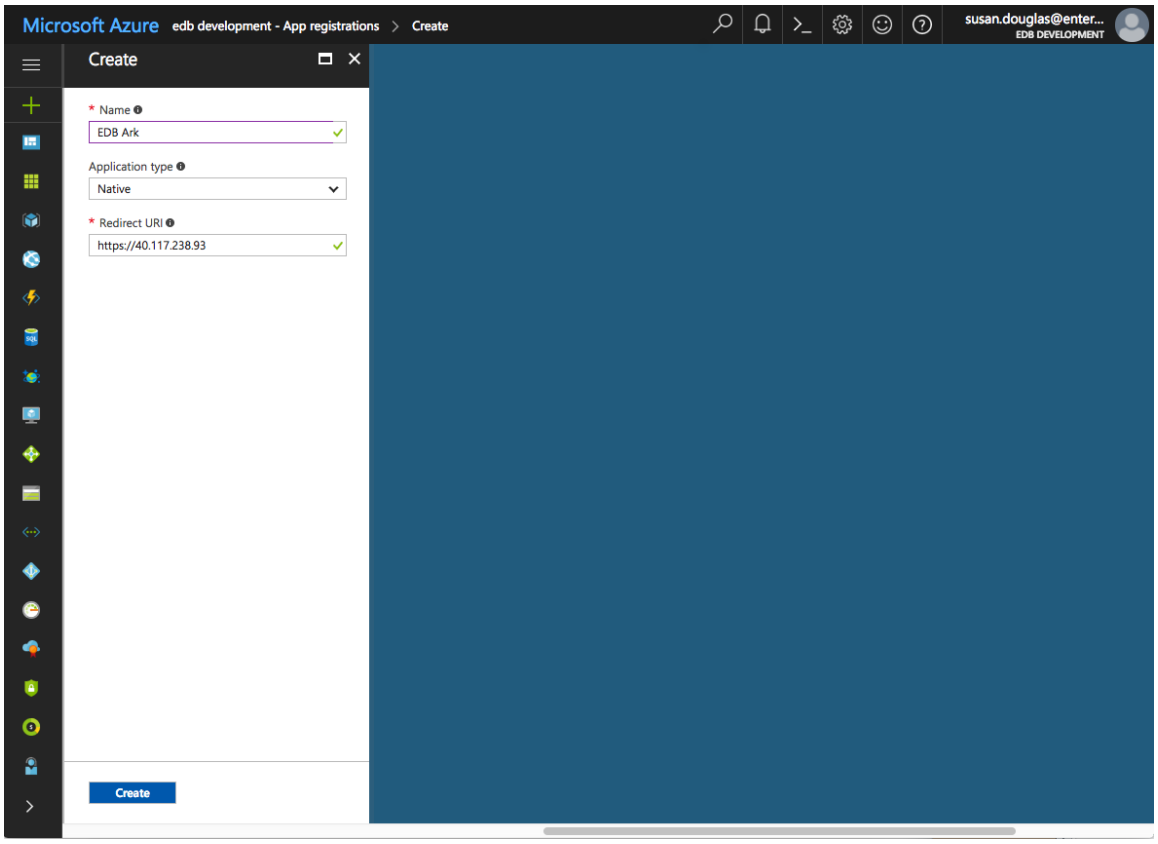

Figure 3.60 – The Create panel.

Use fields on the `Create` panel (see Figure 3.60) 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.

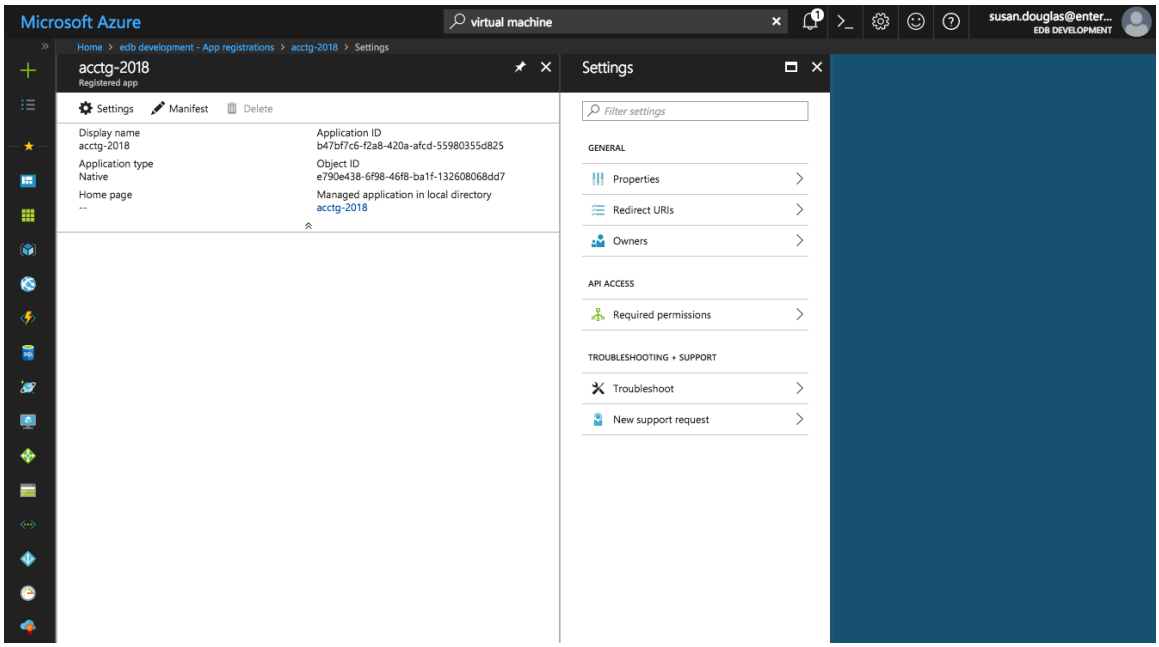

Figure 3.61 – Accessing the Required permissions page.

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. Click the `Settings` icon, and then navigate to the `Required permissions` page for the application, and select `+Add`.

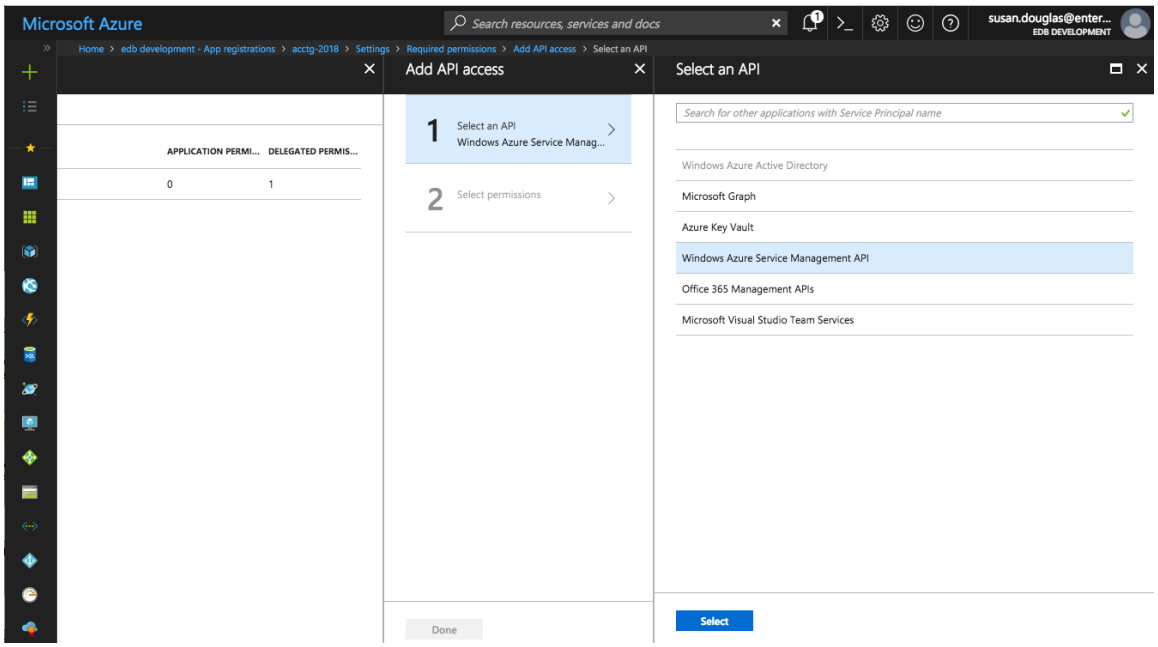

Figure 3.62 – Selecting an API.

Click `Select an API`, and then highlight `Windows Azure Service Management API` (see Figure 3.62).

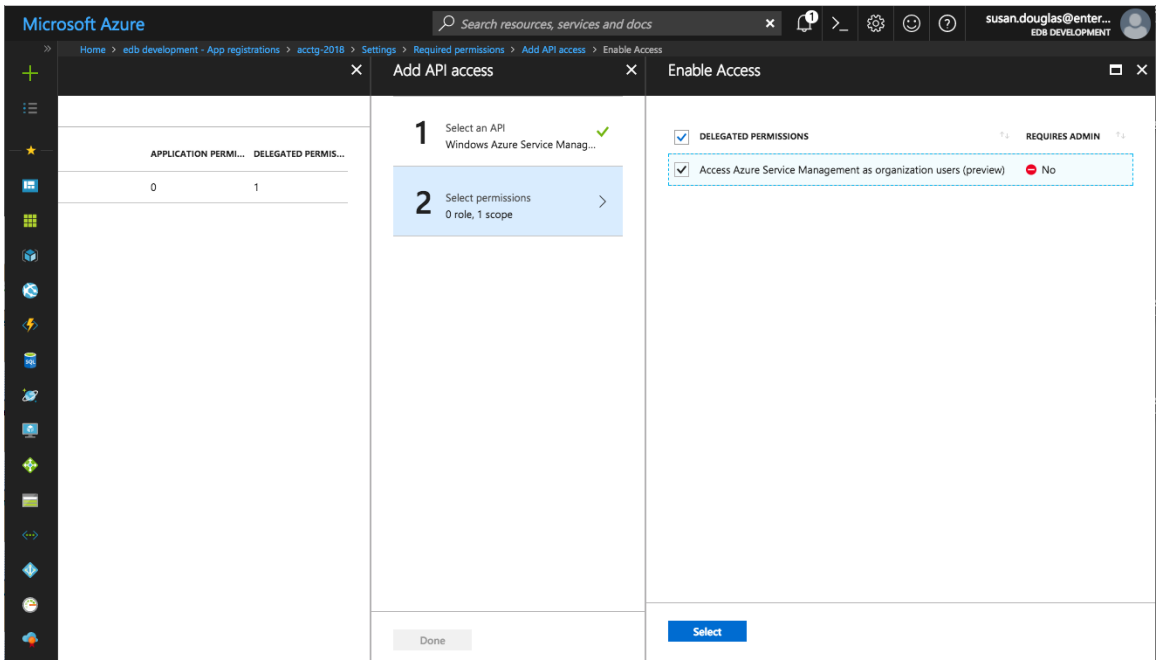

Figure 3.63 – Specifying API permissions.

Click `Select permissions`, and then `Access Azure Service Management` (see Figure 3.63); then, click `Select`.

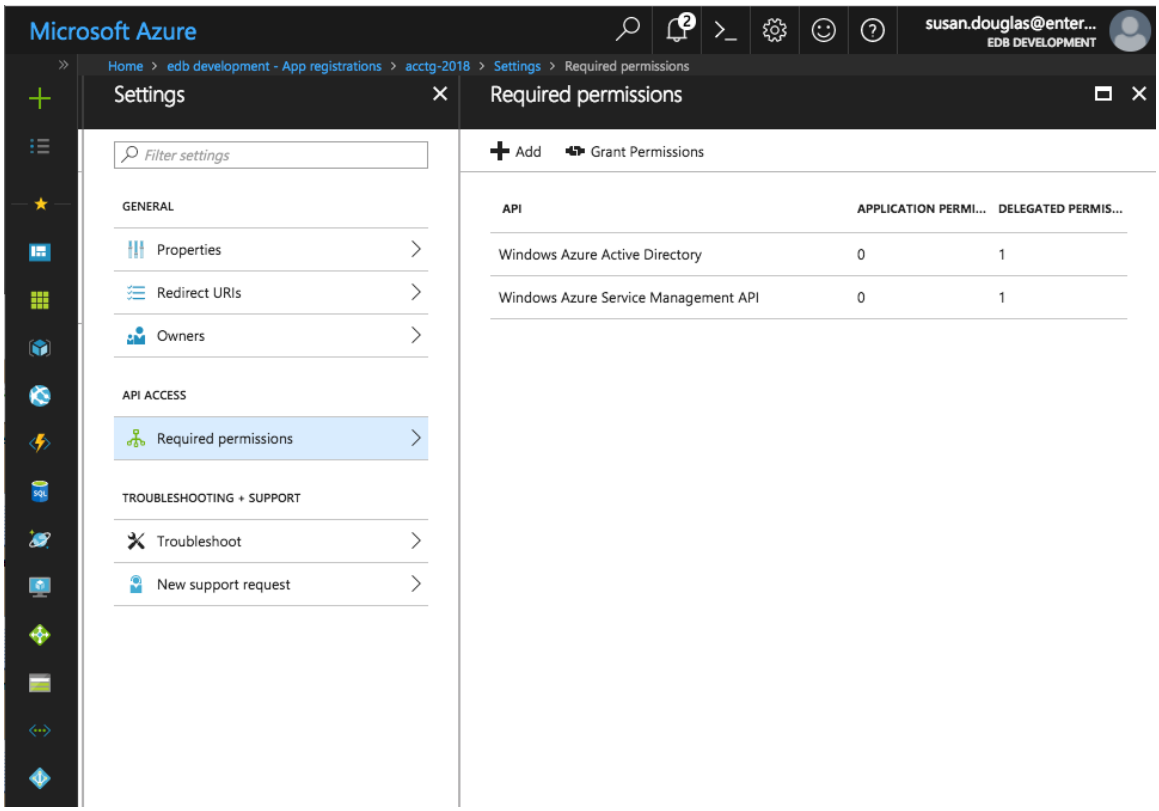

Figure 3.64 – Confirming that the permissions are added.

Then, click `Grant Permissions` (see Figure 3.64).

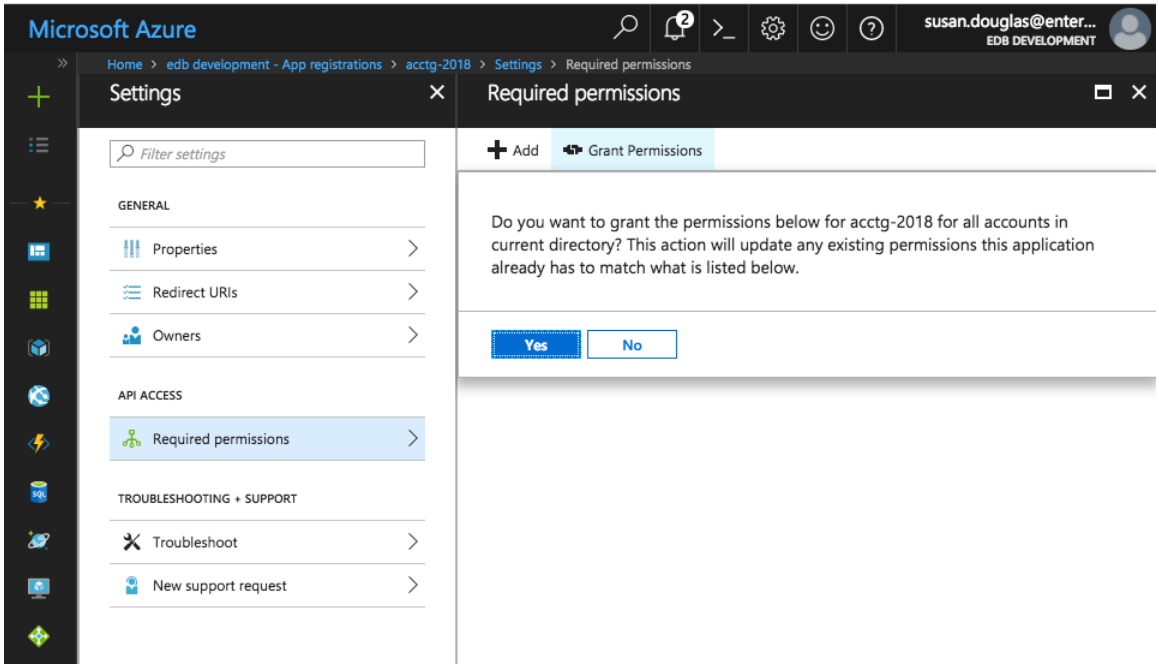

Figure 3.65 – Granting permissions for the Application.

When prompted, click `Yes` to confirm that you wish to grant access permissions (see Figure 3.65).

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

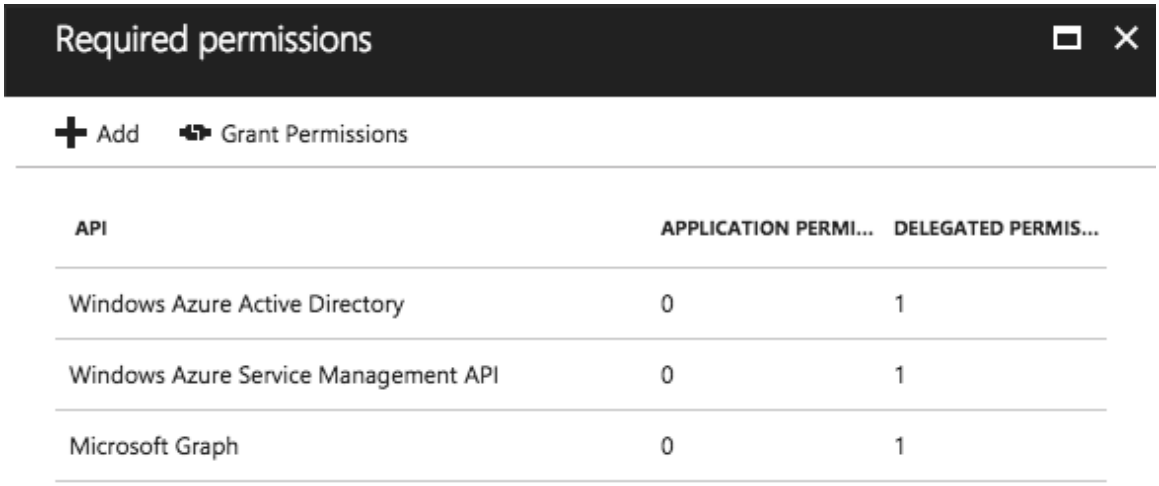

Figure 3.66 – Sufficient Resource permissions.

When you're finished granting permissions, the `Required permissions` list (see Figure 3.66) should include:

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

3.2.5 Configuring the Ark Console

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

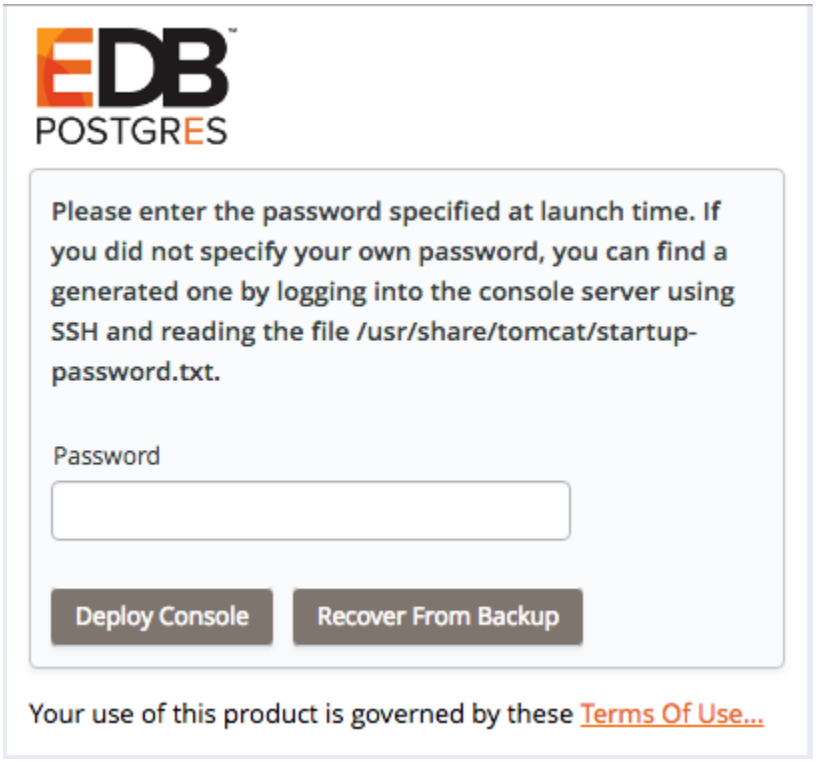

Figure 3.67 – Logging in to the instance.

When prompted, provide the console password (see Figure 3.67). 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 `/usr/share/tomcat/startup-password.txt`.

To retrieve a system assigned password, ssh into the console host and invoke the following command:

```
$ more /usr/share/tomcat/startup-password.txt
h020zdigm95xxqmjonrs
```

The Ark console setup dialog opens as shown in Figure 3.68.

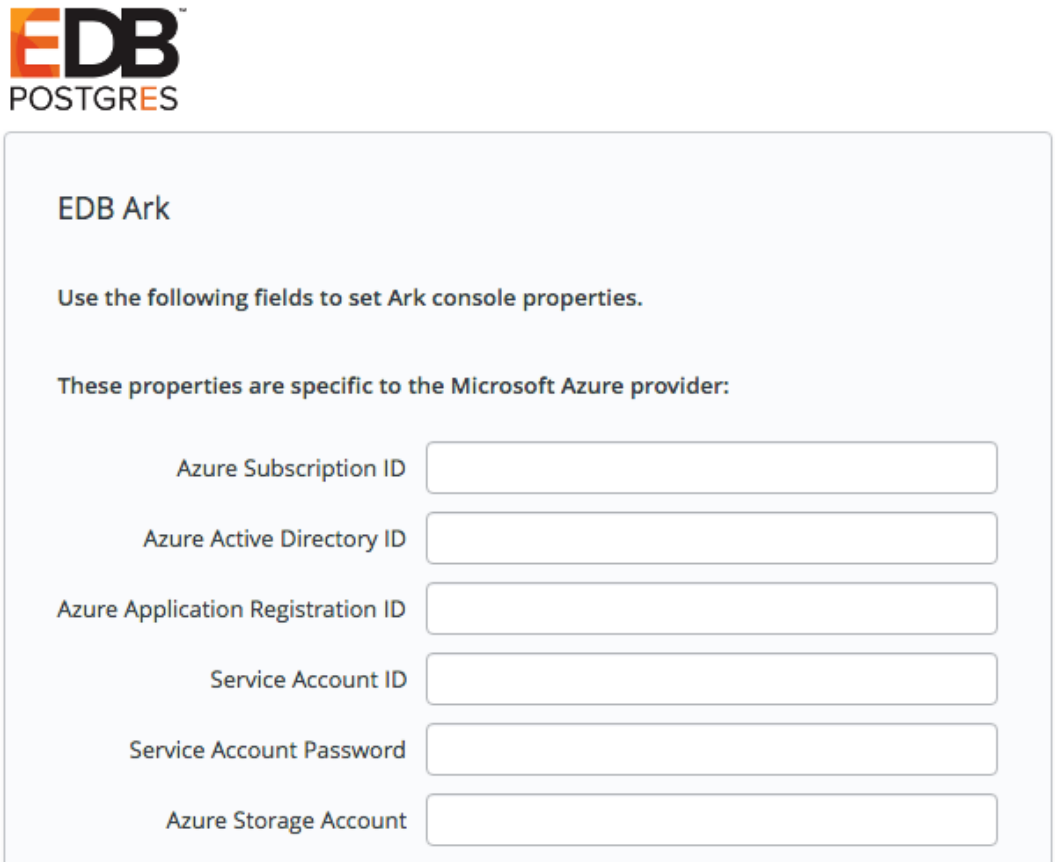

Figure 3.68 – The platform specific property fields.

Use fields in the first portion of the setup dialog to provide platform specific information and configuration details for the Ark console.

- 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.

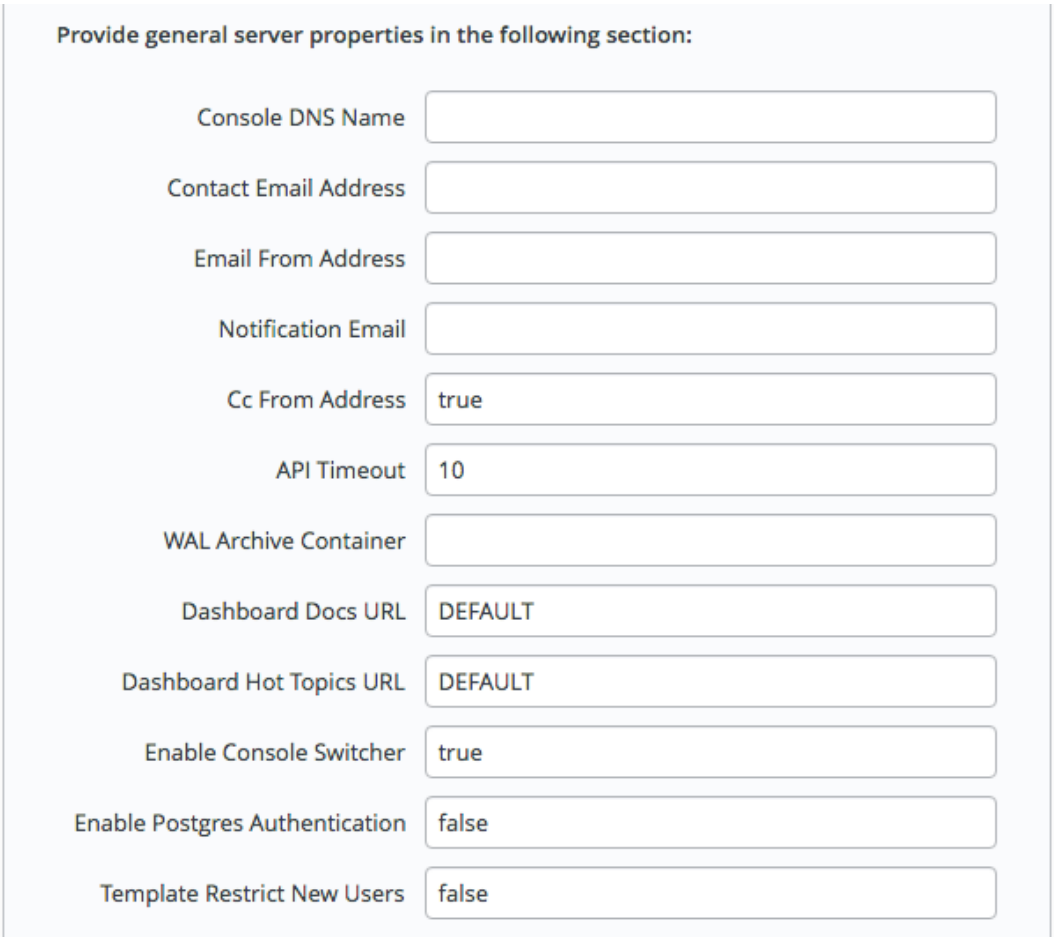

Figure 3.69 – The general server property fields.

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

- Use the `Console DNS Name` field to specify a custom DNS name for the console. The property does not assign the DNS name to the console, but any notification emails that refer to the console will refer to the console by the specified name. If you do not provide a custom DNS name, the IP address of the console will be used in notifications.

- 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.

- Set the `CC From Address` field to `true` to instruct Ark to send a copy of the email to the `Email From Address` anytime a notification email is 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 the `Template Restrict New Users` field to configure the Ark console to make any new user a Template Only user by default. You can later modify the user definition in the `User Administration` table to specify that a user is not a template only user.

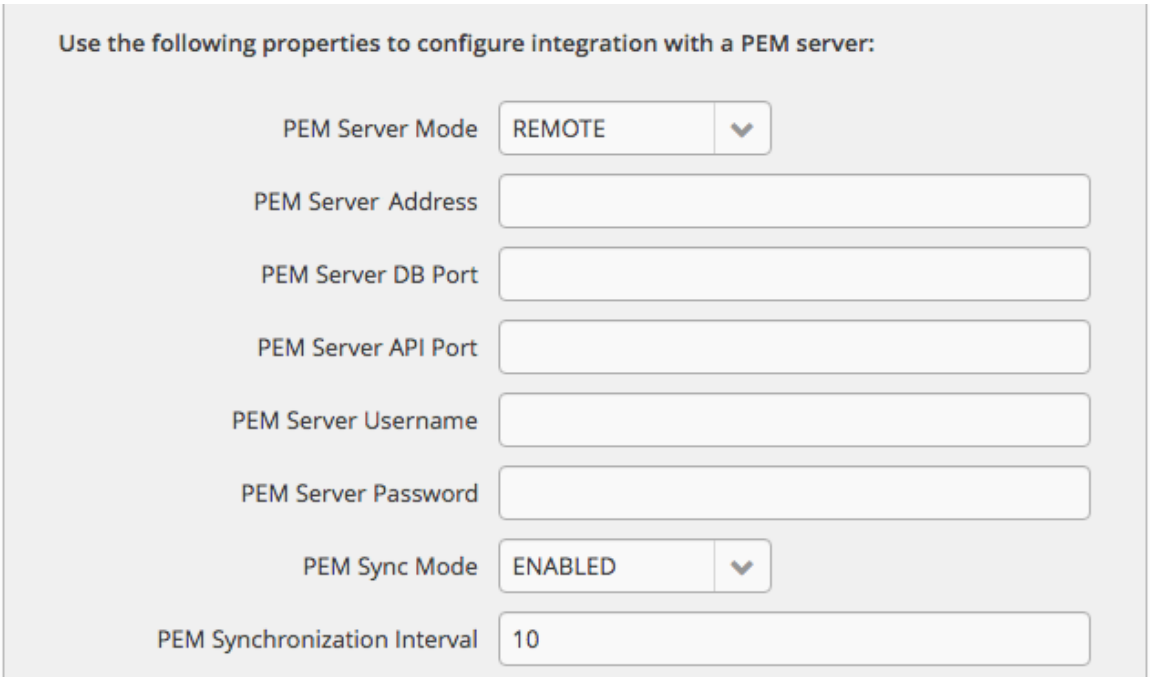

Figure 3.70 – The PEM server property fields.

Use fields in the next section to provide connection details for a PEM server host; this will allow Ark to register and unregister PEM agents and clusters.

- Use the PEM Server Mode drop-down listbox to select a deployment mode:

  Select `DISABLE` to indicate that clusters deployed on the host should not be registered with the PEM server.

  Select `LOCAL` to indicate that you would like to use the PEM server that resides on your local host. If you select `LOCAL`, the PEM deployment will use default values assigned by the installer.

  - The IP address of the PEM server host will be the IP address of the Ark host.
  - The `PEM Server Port` will monitor port `5432`.
  - The PEM server database user will be named `postgres`.

- The password associated with the PEM server will be the same password as the Ark console.

Select `REMOTE` to indicate that you would like to use a PEM server that resides on another host, and provide connection information on the Ark console deployment dialog. If you select `REMOTE`, whenever a new cluster node is created on this console, it will be registered for monitoring by the PEM server.

- Provide the host name or IP address of the PEM server host in the `PEM Server Address` field.
- Specify the port monitored for connections by the PEM server in the `PEM Server DB Port` field.
- Specify the port on the PEM server host used for PEM API connection attempts by the Ark server in the `PEM Server API Port` field. Not valid if the PEM server mode is `DISABLED` or `LOCAL`.
- Provide the name that should be used when authenticating with the PEM server in the `PEM Server Username` field.
- Provide the password associated with the PEM server user in the `PEM Server Password` field.

- Use the `PEM Sync Mode` drop-down listbox to `ENABLE` or `DISABLE` synchronization between the Ark server and the PEM server. For more information about syncing with the PEM server, see Section 2.2.1.
- Use the `PEM Synchronization Interval` field to specify the number of minutes between attempts to synchronize the Ark console with the PEM server.

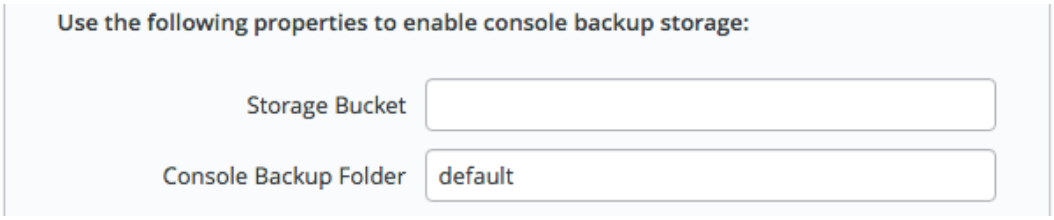

Figure 3.71 – The console backup storage fields.

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.

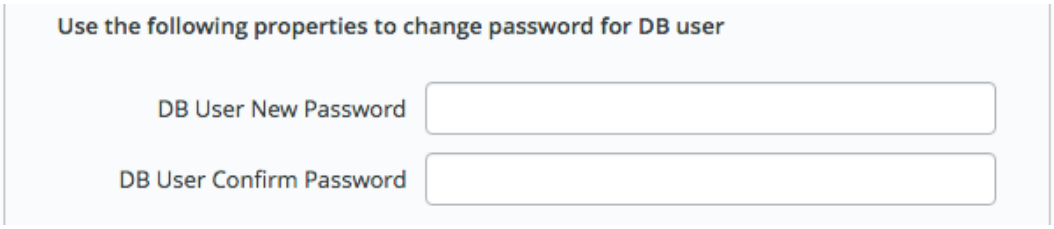

Figure 3.72 - The password fields.

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.

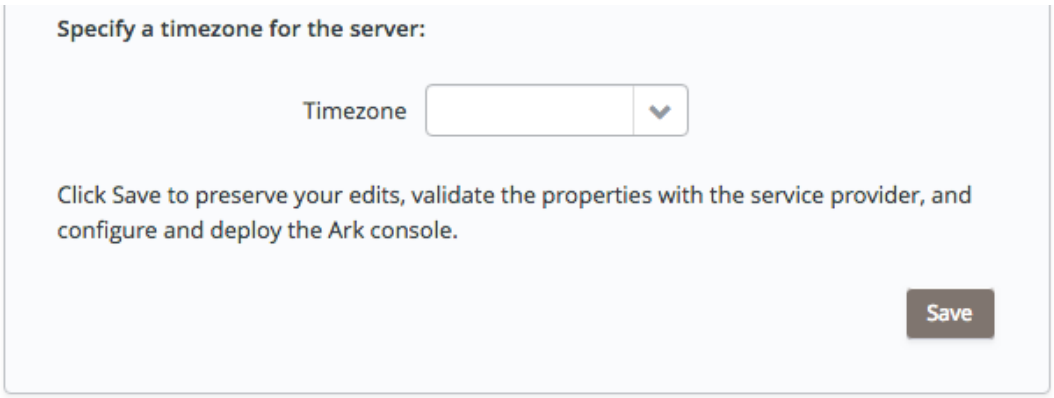

Figure 3.73 - The timezone field.

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 dialog, click the `Save` button to validate and save your preferences; when prompted, click the Restart button to restart the console.

3.2.6 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 3.74).

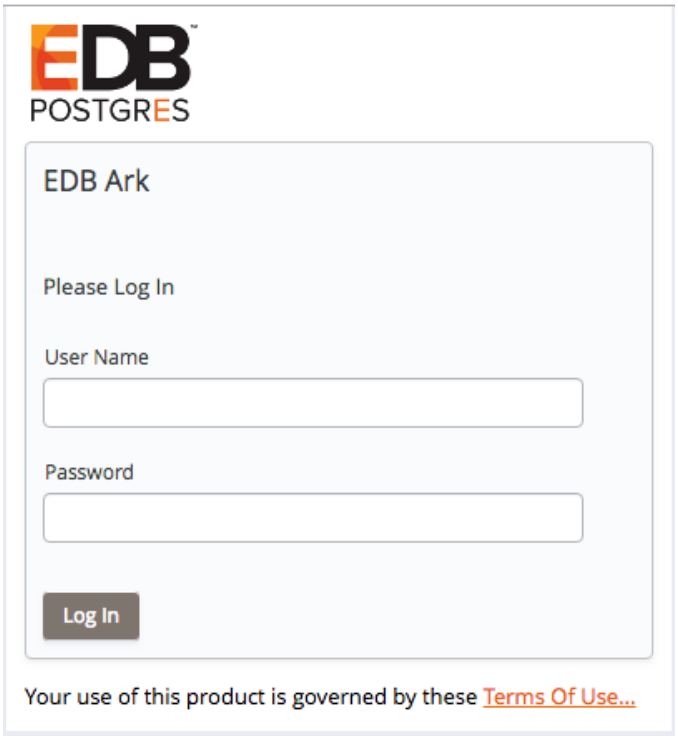

Figure 3.74 - The Login dialog.

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 3.75).

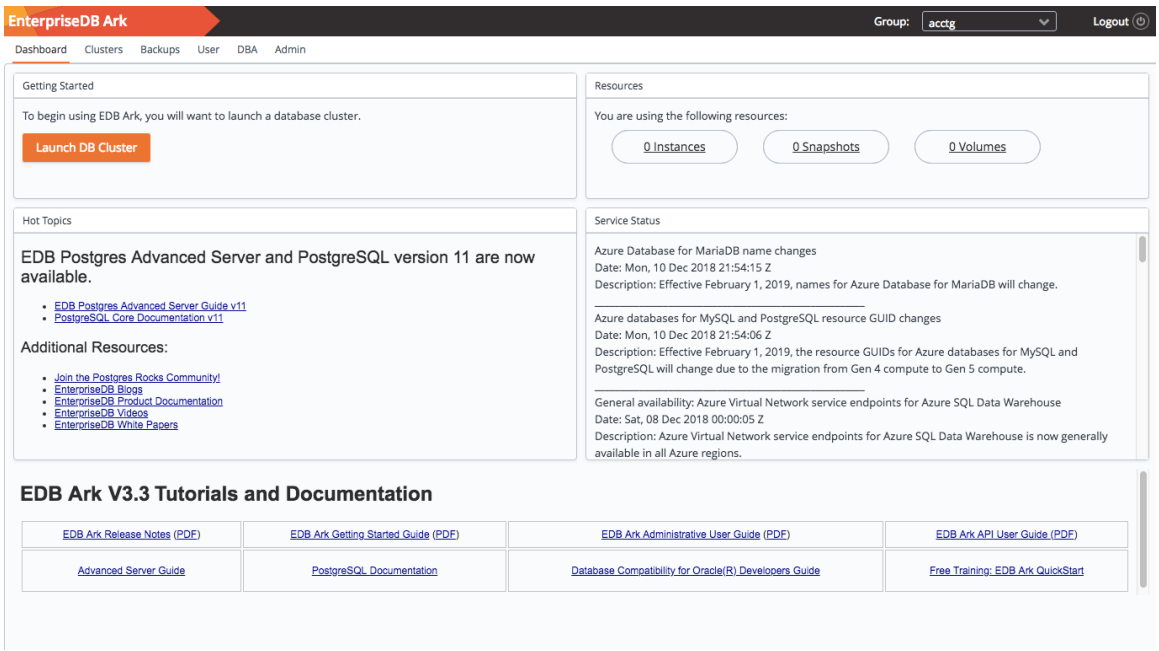

Figure 3.75 - The Dashboard of the EDB Ark Administrator's console.

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).

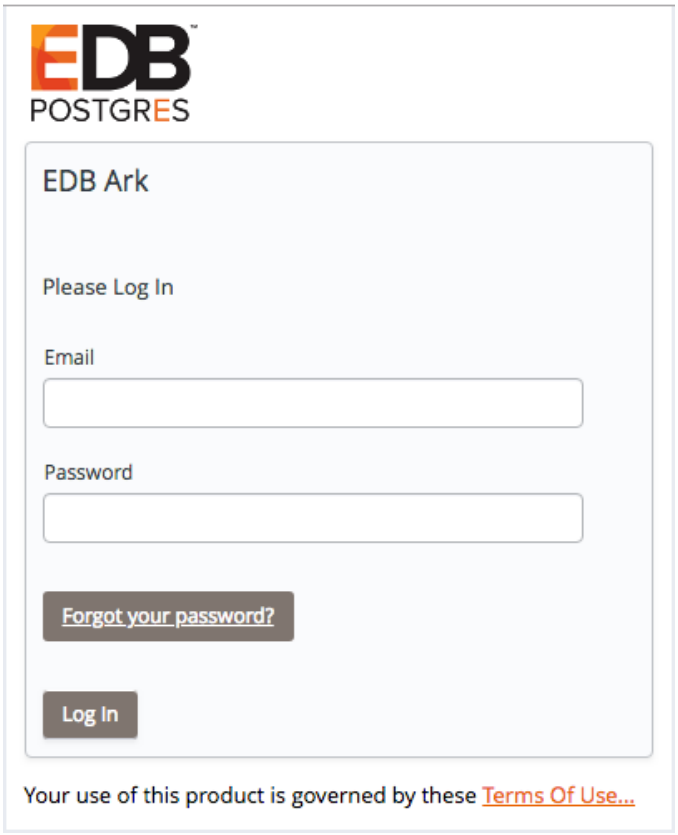

Figure 4.1 - The Login dialog.

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.

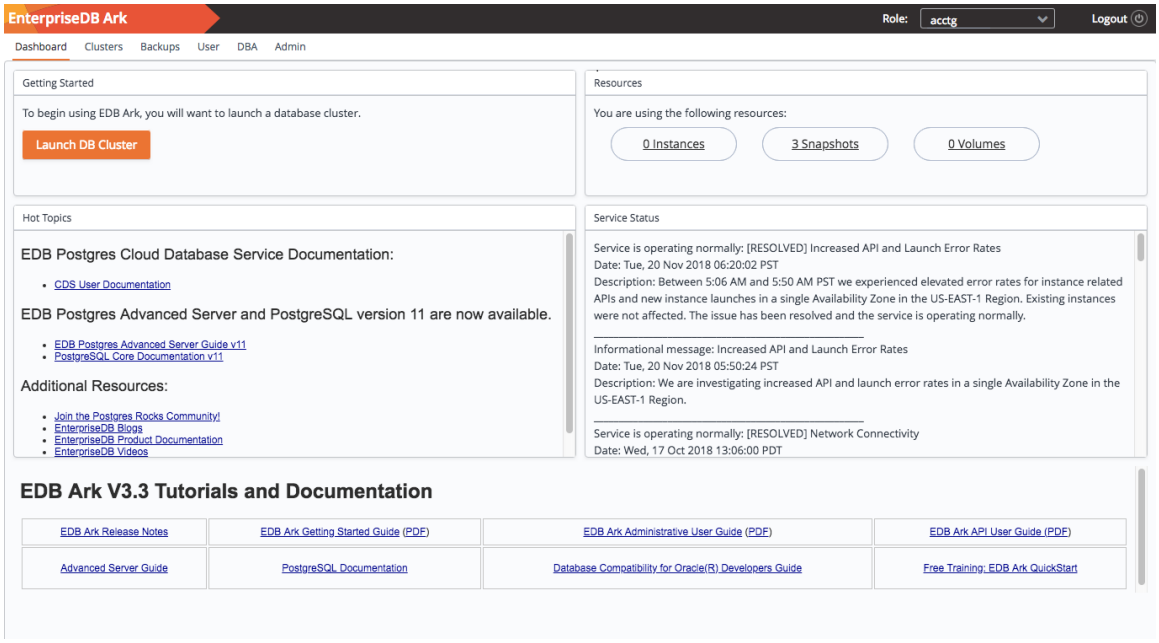

Figure 4.2 - The EDB Ark Administrator's console.

4.1 Using the Admin Tab

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

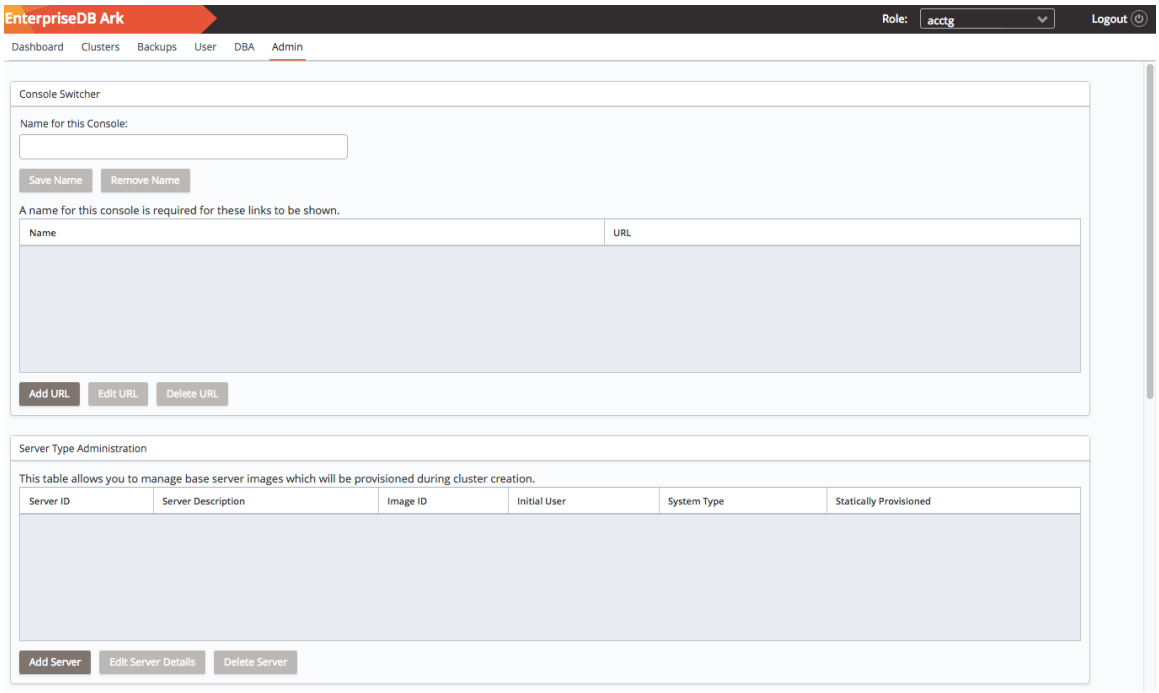

Figure 4.3 – The Admin tab

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, 9.6, 10.0, and 11 (64-bit)
- PostgreSQL 9.4, 9.5, 9.6, 10.0, and 11 (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.

Template Administration

Use the Template Administration section to create and manage database cluster templates. A template contains a predefined set of server options that determine the configuration of a cluster.

An administrator can use a template to:

- simplify creation of clusters that use a common configuration.

- predefine supported cluster configurations.

- limit access to server resources for a *Template Only* user. A Template Only user must use a template when deploying a new cluster.

For more information about creating and using a template, see Section 4.1.4.

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 4.1.5.

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 4.1.6.

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.
- Specify that a user must use a template when deploying a cluster.

For information about user administration options, see Section 4.1.7.

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 4.1.8.

Backup Ark Console

Click the `Backup Now` button to start a console backup. A popup will confirm that you have requested a manual backup of the console.

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 4.1.10.

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.

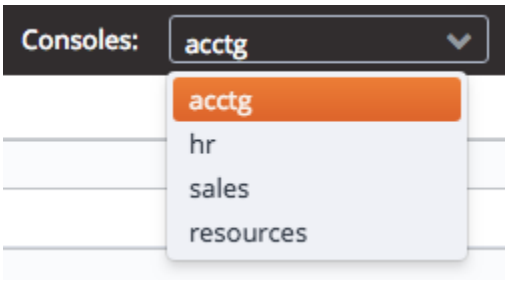

Figure 4.4 – The Consoles drop-down.

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).

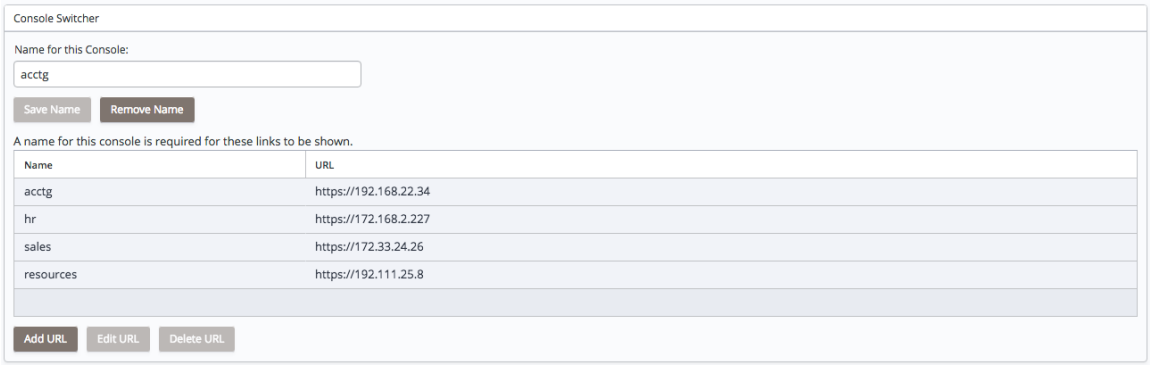

Figure 4.5 – The Console Switcher section of the Admin tab.

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. 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).

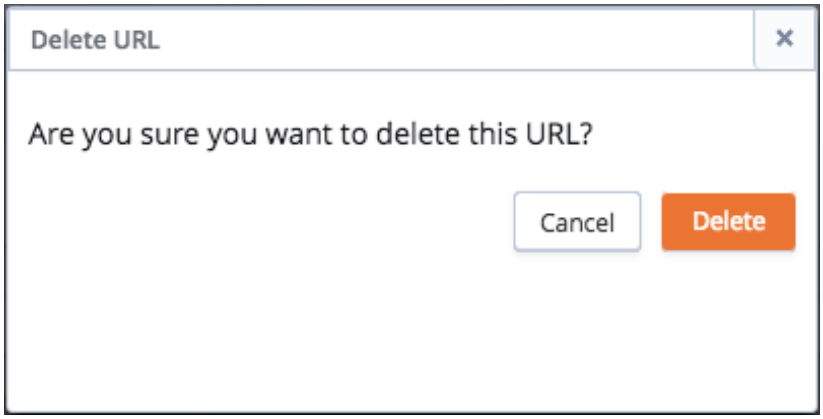

Figure 4.9 – The Edit URL dialog.

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).

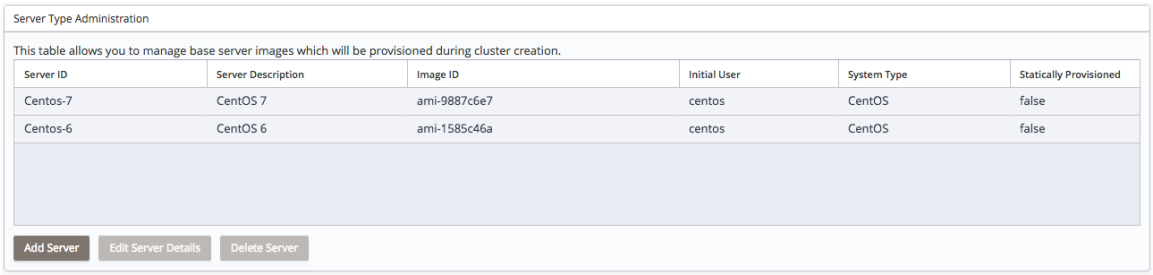

Figure 4.10 – The Server Type Administration section of the Admin tab.

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.

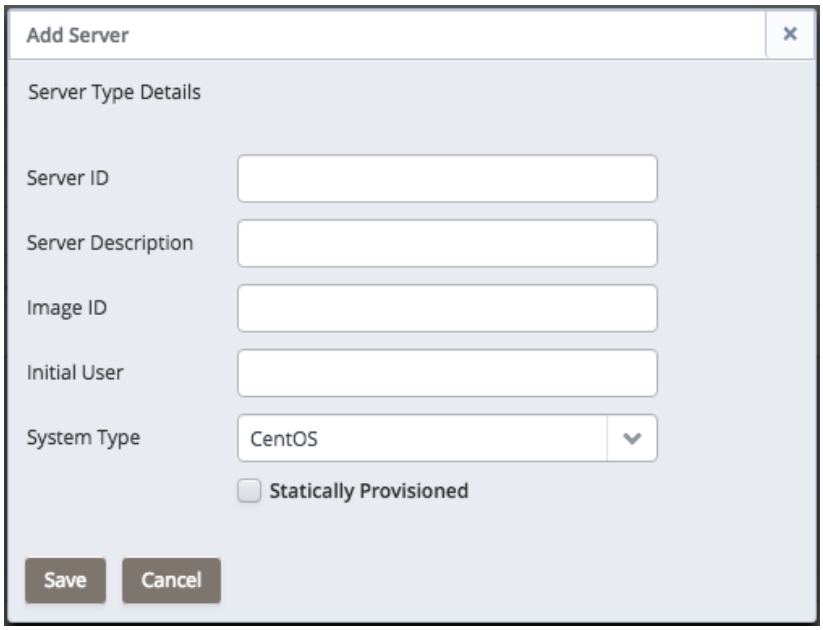

Figure 4.11 – The Add Server dialog.

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 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-RAW
  - OpenLogic:CentOS:7.5
  - 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 11.

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.

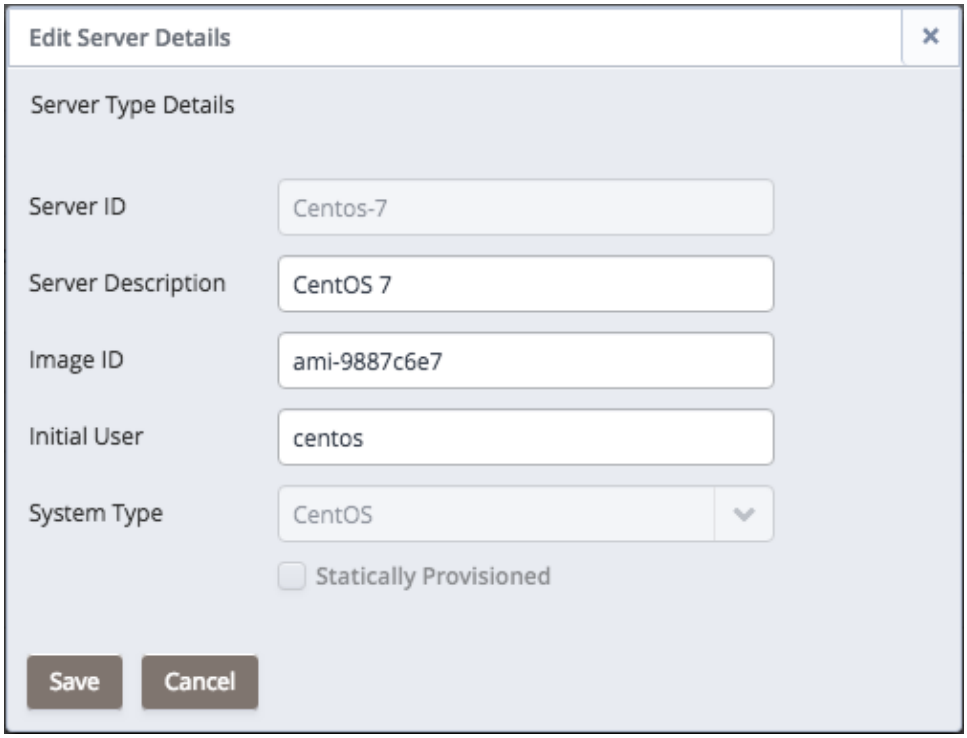

Figure 4.12 – The Edit Server dialog.

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).

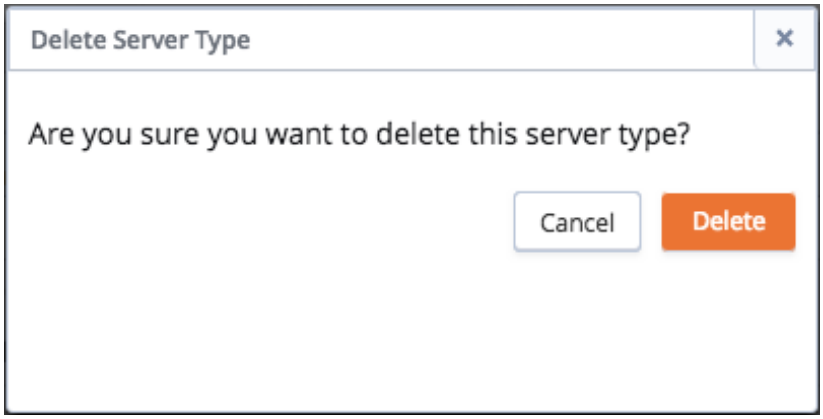

Figure 4.13 – The Delete Server Type dialog.

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

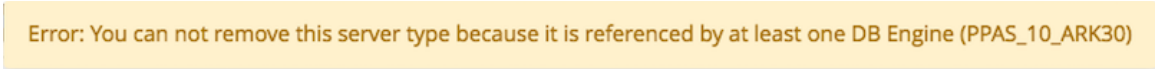

Figure 4.14 – You cannot remove a server with dependencies.

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).

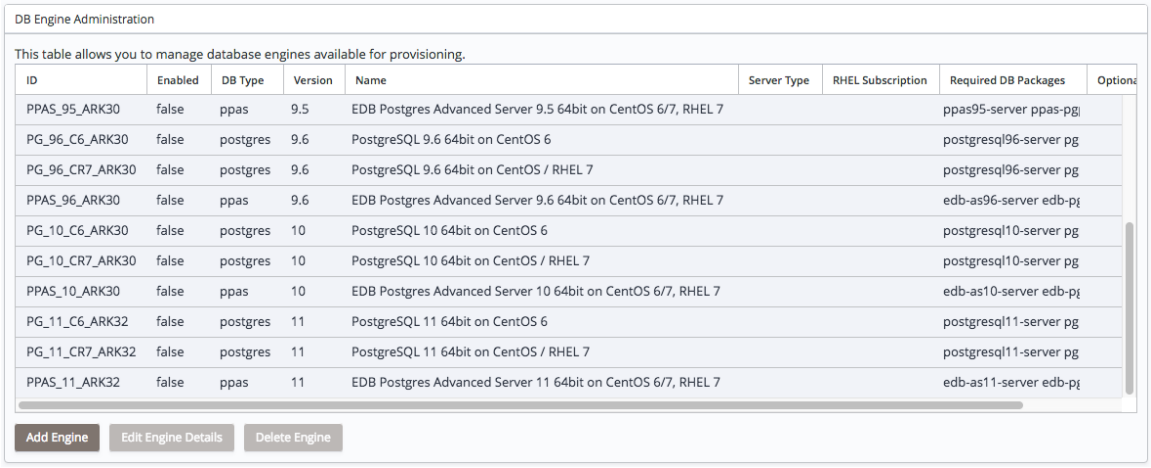

Figure 4.15 – The DB Engine Administration section of the Admin tab.

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 9.4 64bit on CentOS 6

Repository Location:
`https://yum.postgresql.org/9.4/redhat/rhel-6-x86_64/pgdg-redhat-repo-latest.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:
`https://yum.postgresql.org/9.4/redhat/rhel-7-x86_64/pgdg-redhat-repo-latest.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:
`https://yum.postgresql.org/9.5/redhat/rhel-6-x86_64/pgdg-redhat-repo-latest.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:
`https://yum.postgresql.org/9.5/redhat/rhel-7-x86_64/pgdg-redhat-repo-latest.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:
`https://yum.postgresql.org/9.6/redhat/rhel-6-x86_64/pgdg-redhat-repo-latest.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:
`https://yum.postgresql.org/9.6/redhat/rhel-7-x86_64/pgdg-redhat-repo-latest.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-redhat-repo-latest.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-redhat-repo-latest.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:
`https://USERNAME:PASSWORD@yum.enterprisedb.com/tools/redhat/rhel-\$releasever-\$basearch`
`https://USERNAME:PASSWORD@yum.enterprisedb.com/dependencies/redhat/rhel-\$releasever-\$basearch`
`https://yum.postgresql.org/11/redhat/rhel-6-x86_64/pgdg-redhat11-11-2.noarch.rpm`

Required Packages: `postgresql11-server pgpool-II-11 pgpool-II-11-extensions edb-pem-agent`

PostgreSQL 11 64bit on CentOS 6

Repository Locations:
`https://yum.postgresql.org/11/redhat/rhel-6-x86_64/pgdg-redhat-repo-latest.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 11 64bit on CentOS / RHEL 7

Repository Locations:

```
https://yum.postgresql.org/11/redhat/rhel-7-x86_64/pgdg-
redhat-repo-latest.noarch.rpm
https://yum.postgresql.org/11/redhat/rhel-7-x86_64/pgdg-
redhat11-11-2.noarch.rpm
https://USERNAME:PASSWORD@yum.enterprisedb.com/dependencies
/redhat/rhel-\$releasever-\$basearch
```

Required Packages: `postgresql11-server pgpool-II-11 pgpool-II-11-extensions edb-pem-agent`

EDB Postgres Advanced Server 11 64bit on CentOS 6/7, RHEL 7

Repository Locations:

```
http://USERNAME:PASSWORD@yum.enterprisedb.com/10/redhat/rhe
l-$releasever-$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`.

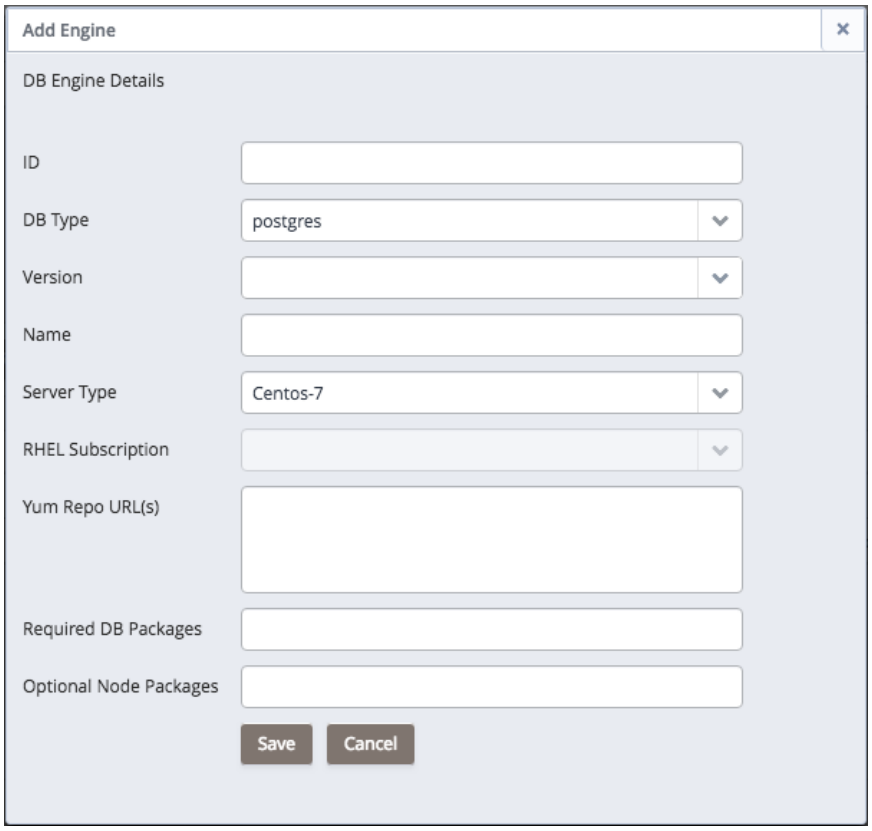

Figure 4.16 – The Add Engine dialog.

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 an Advanced Server 11 database, the package list must include:

```
edb-as11-server
edb-pgpool37
edb-as11-pgpool37-extensions
edb-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
```

For a PostgreSQL 11 database, the package list must include:

```
postgresql11-server
pgpool-II-11
pgpool-II-11-extensions
edb-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).

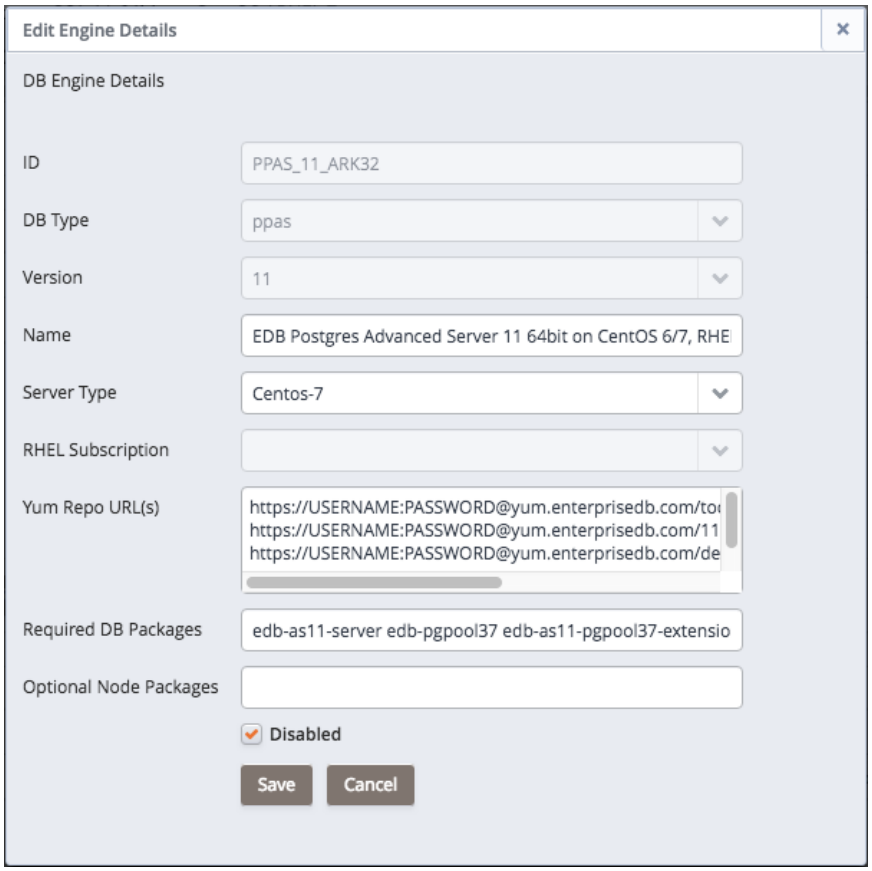

Figure 4.17 – The Edit Engine Details dialog.

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).

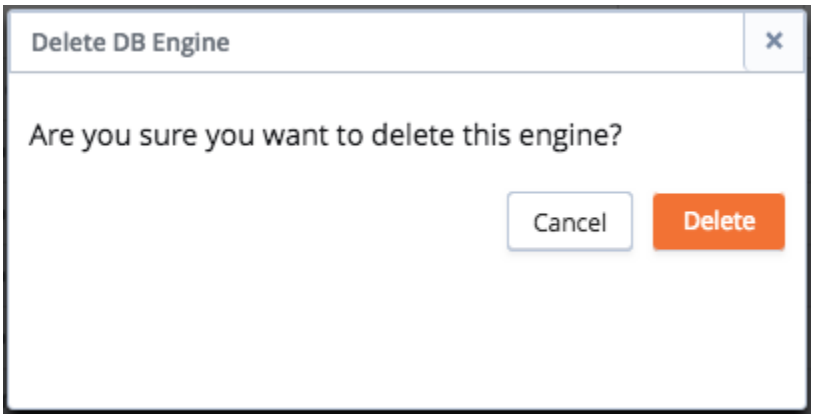

Figure 4.18 – The Delete DB Engine dialog.

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

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 a warning message.

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.

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`.

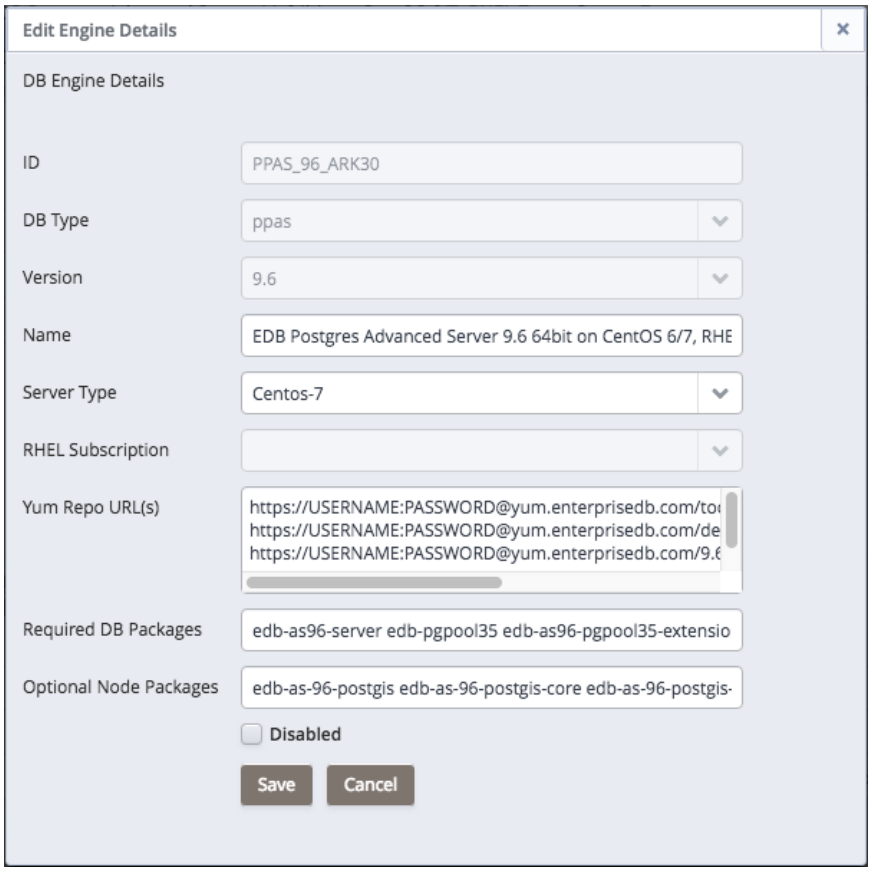

Figure 4.28 – Modifying the Engine Details dialog.

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
  ```

  The PostGIS installation packages for Advanced Server 11 are:

  ```
  edb-as-11-postgis
  edb-as-11-postgis-core
  edb-as-11-postgis-docs
  edb-as-11-postgis-jdbc
  edb-as-11-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.

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 Template Administration

A template contains a predefined set of server options that determine the configuration of a database cluster. A template can simplify creation of clusters that use a common configuration, or limit user access to costly resources such as large server classes. Use functionality offered in the `Template Administration` section (see Figure 4.30) of the `Admin` tab to create and manage templates.

Figure 4.30 – The Template Administration section of the Admin dashboard.

Use the `TEMPLATES ONLY` column of the `User Administration` table to specify that a user must use a template. A *Template Only* user will have access to only those templates that specify a role or tenant in which they have membership in the `Select Roles` section of the `Add Template` dialog.

If a user is specified as a Template Only user:

- They must use a template when deploying a cluster.
- They will be restricted to the scaling policies defined in the template.
- They cannot modify a manually-defined cluster created by another user.
- They can only create clusters in a server class that exists in an available template.
- They must use a template when cloning or restoring from backup.
- They may only delete backups of template created clusters.
- They may not delete last backup of a template created cluster if the cluster had been deleted (removing the last artifact of any cluster).

To create a template, click the `Add Template` button; the `Add Template` dialog opens (see Figure 4.31).

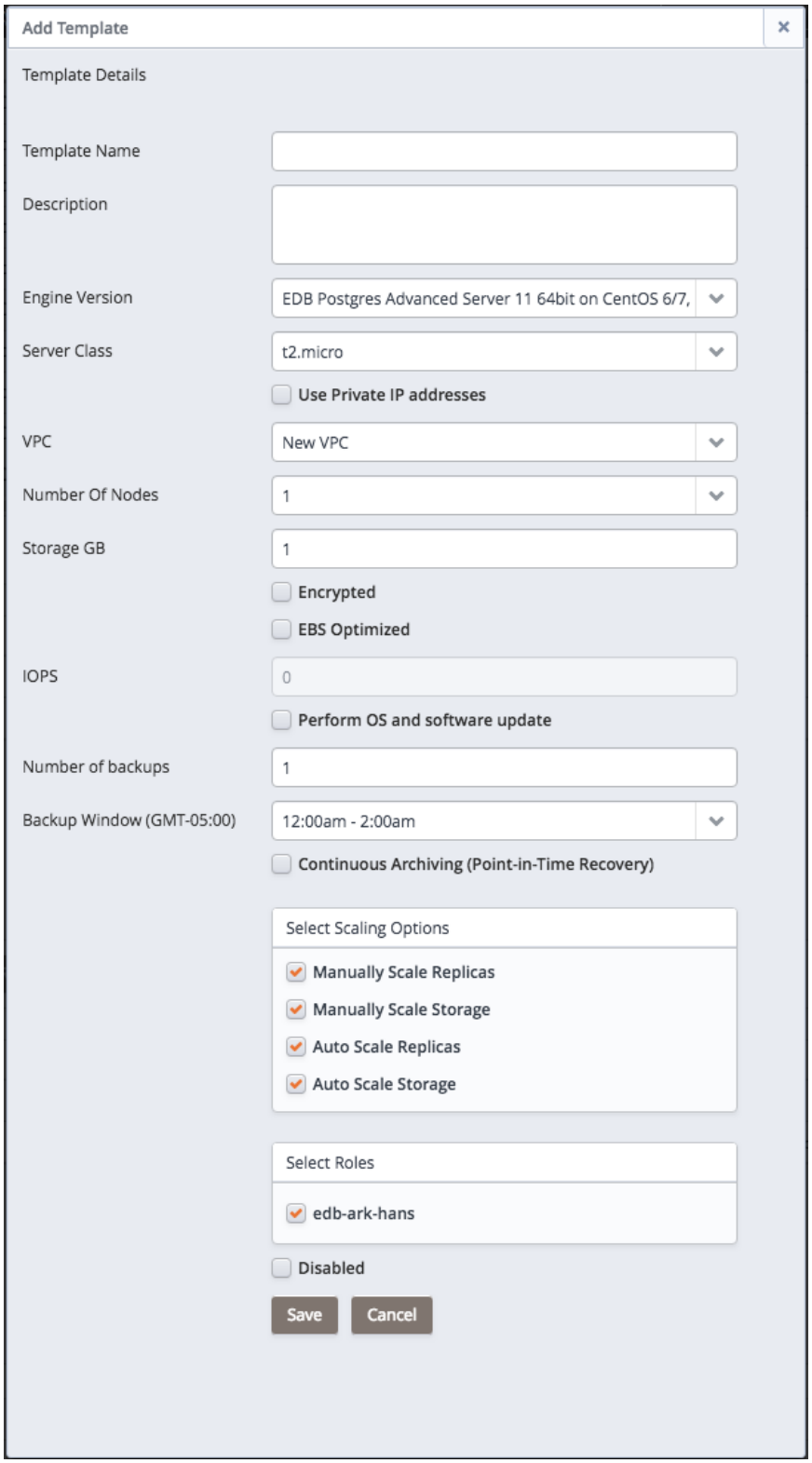

Figure 4.31 – The Template Administration section of the Admin dashboard.

Use fields on the Add Template dialog to define a new template:

- Provide a user-friendly name for the template in the Template Name field.
- Use the Description field to provide a description of the template.
- Use the drop-down listbox in the Engine Version field to select the version of the Postgres engine that you wish to use on clusters configured by the template.
- Use the drop-down listbox in the Server Class field to specify the size of each cluster node. The server class determines the size and type (compute power and RAM) of any cluster configured by the template.
- If your cluster resides on an Amazon AMI, use the drop-down listbox in the VPC field to specify the identity of the network in which clusters configured by the template should reside.
- Use the drop-down listbox in the Number of nodes field to specify the number of nodes that should be created in each cluster.
- Use the Storage GB field to specify the initial size of the data space (in Gigabytes).
- Check the box next to Encrypted to indicate that the cluster should be encrypted. EDB Ark uses the aes-xts-plain (512-bit) cipher suite to provide an encryption environment that is both secure and transparent to connecting clients. When encryption is enabled, everything residing on the cluster is encrypted except for the root filesystem.
- If the cluster will reside on an AWS host, check the box next to EBS Optimized to specify that the cluster should use an Amazon EBS-optimized instance and provisioned IOPS to guarantee a level of I/O performance.
- The IOPS field is enabled for those clusters that will reside on an EBS-optimized instance. If applicable, specify the level of I/O performance that will be maintained for the cluster by automatic scaling. The maximum value is 30 times the size of your cluster; for example, if you have a 4 Gigabyte cluster, you can specify a maximum value of 120.
- Check the box next to Perform OS and Software update to specify that a software update should be performed whenever the cluster is provisioned. Please note: this option is disabled if the cluster uses a statically provisioned server.

- Use the Number of backups field to specify the number of backups that will be retained for the cluster. When the specified number of server backups is reached, EDB Ark will delete the oldest backup to make room for a new backup.

- Use the Backup Window field to specify a time that it is convenient to perform a cluster backup.

- Check the box next to Continuous Archiving (Point-in-Time Recovery) to enable point-in-time recovery for the cluster. When enabled, a base backup is automatically performed that can to be used to restore to a specific point in time. All subsequent automatic scheduled backups will also support point-in-time recovery.

- Check the boxes next to the options in the Select Scaling Options box to indicate which options will be available to template users. Check the box next to:

    - Manually Scale Replicas to specify that users of this template will be allowed to manually scale replica nodes configured by this template.

    - Manually Scale Storage to specify that users of this template will be allowed to manually scale storage on clusters configured by this template.

    - Auto Scale Replicas to specify that users of this template will be able to configure automatic node scaling for clusters configured by this template.

    - Auto Scale Storage to specify that users of this template will be able to configure automatic storage scaling for clusters configured by this template.

- Check the box to the left of a role or tenant name in the Select Roles or Select Tenants box to indicate that members of the selected role or tenant can use the template.

When you've completed the Add Template dialog, click Save to create the defined template; click Cancel to close the dialog and exit without saving your work.

If you select Launch From Template on the Create a New Server Cluster dialog, you will be prompted to select the template you wish to use from the Template Name drop-down listbox. After selecting a template, you can use the Full Template Details link to open a popup that displays detailed information about the configuration of clusters deployed with the template (see Figure 4.32).

Figure 4.32 – The template details popup.

4.1.5 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 4.33).

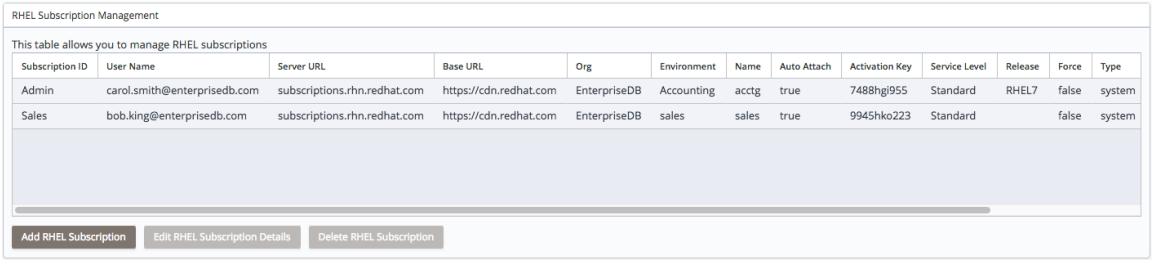

Figure 4.33 – the RHEL Subscription Management section.

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

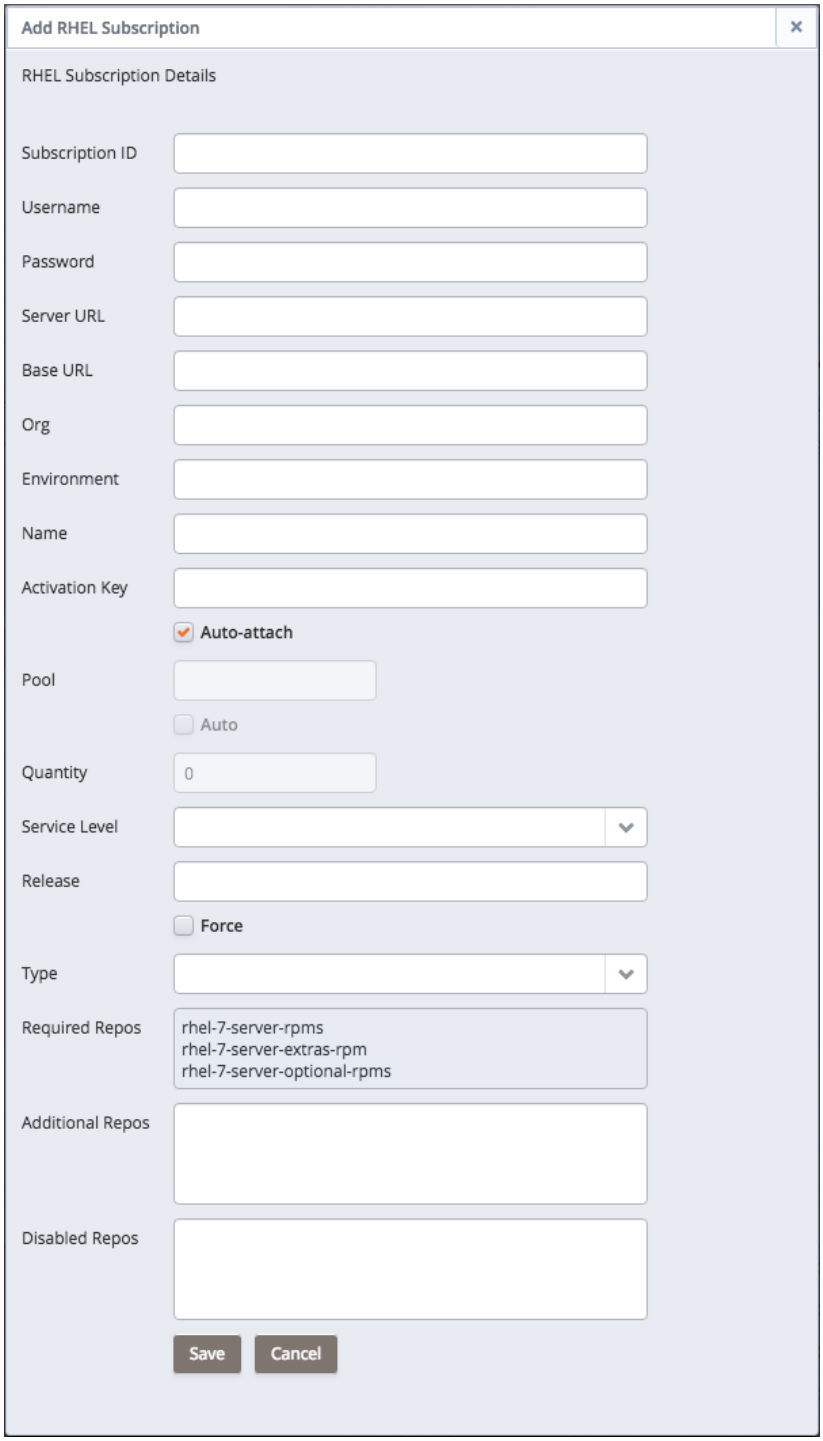

Figure 4.34 – The Add RHEL Subscription dialog.

Use fields on the `Add RHEL Subscription` dialog (see Figure 4.34) 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 4.35).

Figure 4.35 – 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.

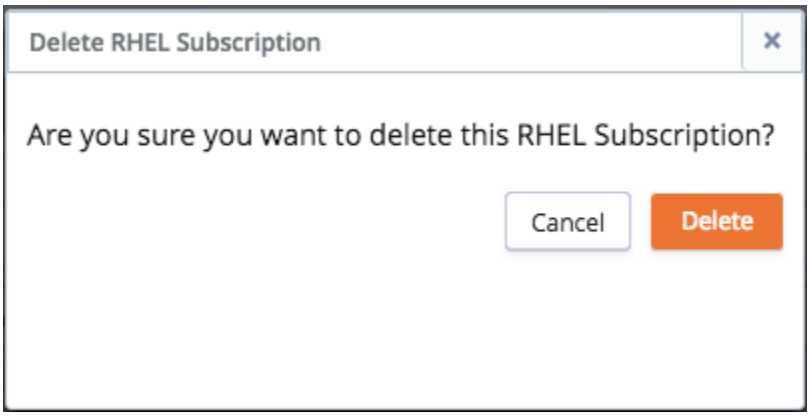

Figure 4.36 – Confirming that you wish to delete a subscription description.

Click the `Delete` button to confirm that you wish to delete the subscription definition, or `Cancel` to exit without deleting the definition (see Figure 4.36).

4.1.6 Managing Amazon Roles

Amazon Role ARNs that are listed in the `IAM Roles Administration` table (see Figure 4.37) 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.

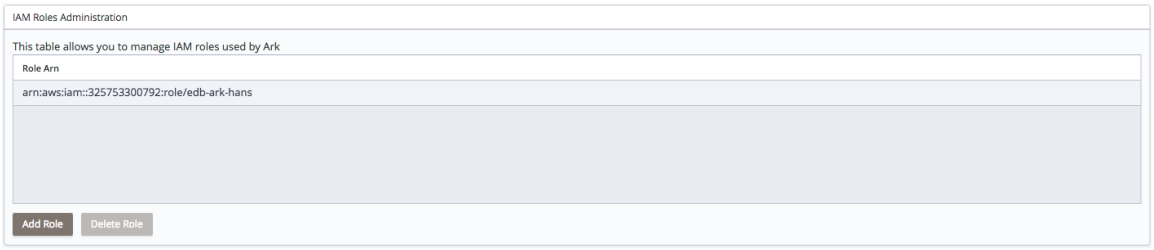

Figure 4.37 – The Roles Administration dialog.

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 4.38).

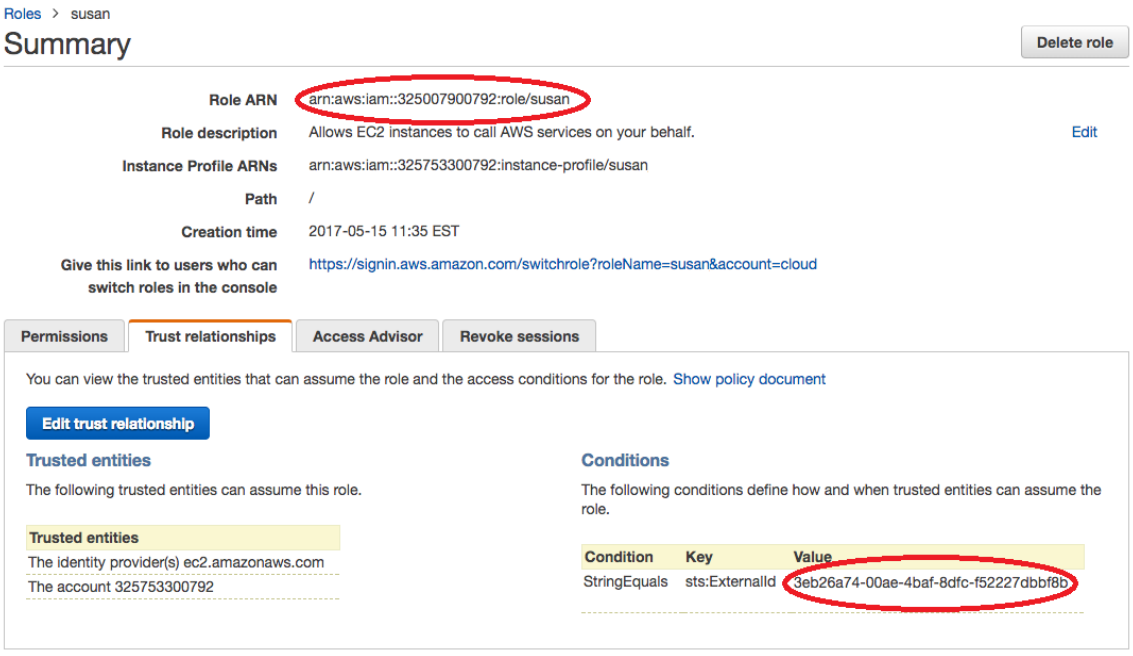

Figure 4.38 – The Roles Administration dialog.

To add a Role ARN to the table, click the Add Role button; the Add Role dialog opens as shown in Figure 4.39.

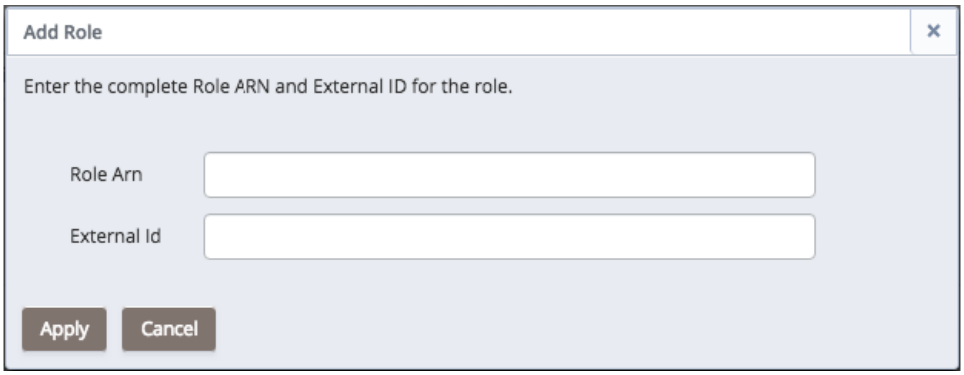

Figure 4.39 – The Roles Administration dialog.

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.

4.1.7 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.

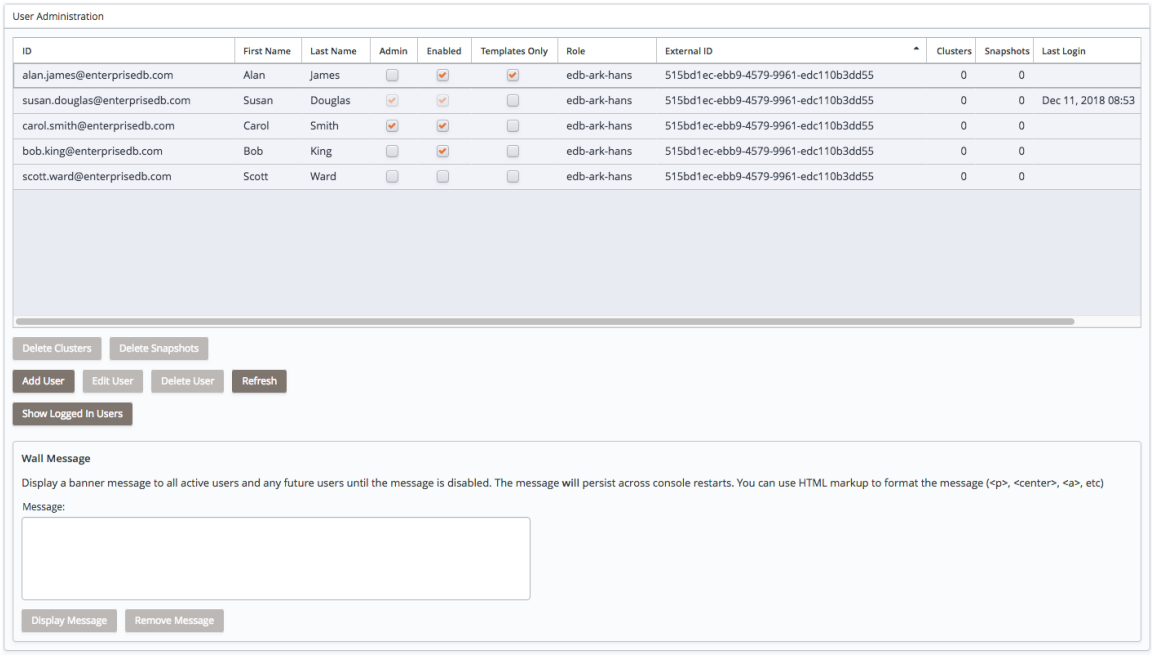

Figure 4.40 – User administration features of the Amazon console.

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.
- specify that a user must use a template when deploying a cluster.

Adding a User

If available for your configuration, you can click the `Add User` button to access the `Add User` dialog (see Figure 4.41) and register a new user account for the Ark console.

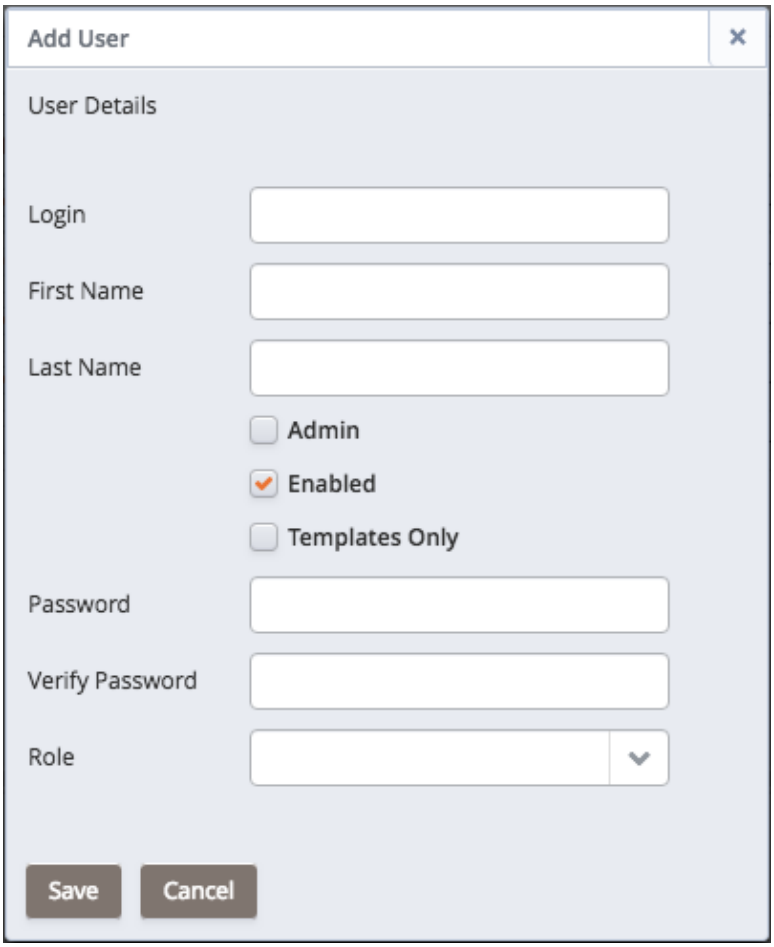

Figure 4.41 – The Add User dialog on an Amazon host.

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.
- Check the box next to `Templates Only` to specify that a user must use a template when deploying a cluster.

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

After making changes to modifiable fields, click the `Save` button to make the changes persistent. Click `Cancel` to exit without saving any changes.

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 4.43). Please note that you 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.

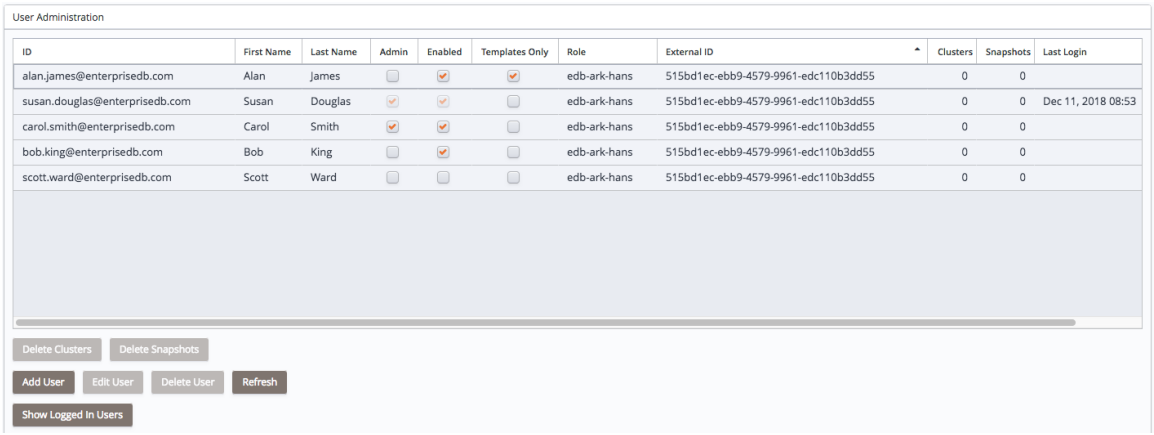

Figure 4.43 – The user table of an AWS console

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.
- 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.
- Check the box in the `TEMPLATES ONLY` column to indicate that the user must use templates when deploying clusters.

- 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.

After adding the user to the Ark console, use the psql client application to add the user to the backing postgres database. To use the psql client, SSH to the host of the Ark console. Then, 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.

Please note: 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 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.

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 4.44). Please note that you must use a client application to connect to the Ark console and add the user to the `postgres` database before the user is added to the table or allowed to connect.

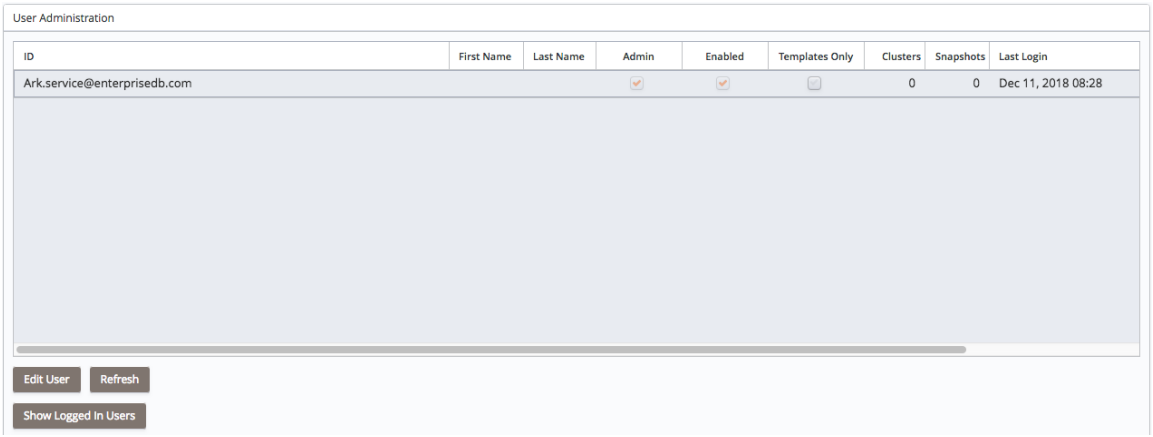

Figure 4.44 –User administration table on the Azure console.

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.
- Check the box in the `TEMPLATES ONLY` column to indicate that the user must use templates when deploying clusters.

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

4.1.3.5 Displaying Connected Users

Click the `Show logged in users` button to display the `Logged in users` dialog (see Figure 4.46).

Figure 4.46 – 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.

4.1.3.6 Managing the Wall Message

Provide a message in the `Message` field (shown in Figure 4.47) 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.

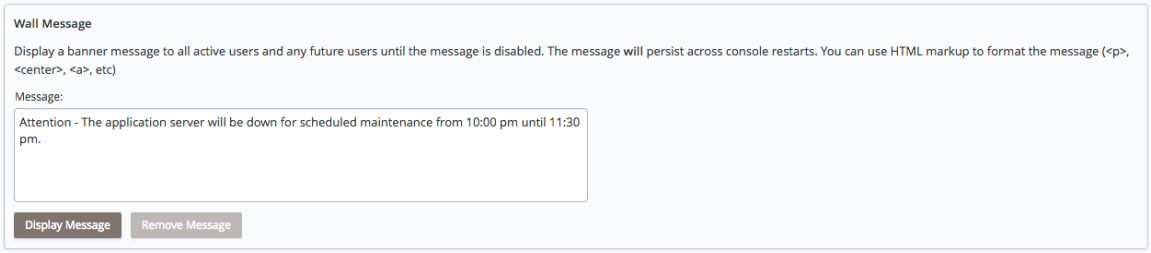

Figure 4.47 - Modifying the Wall Message.

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 4.48).

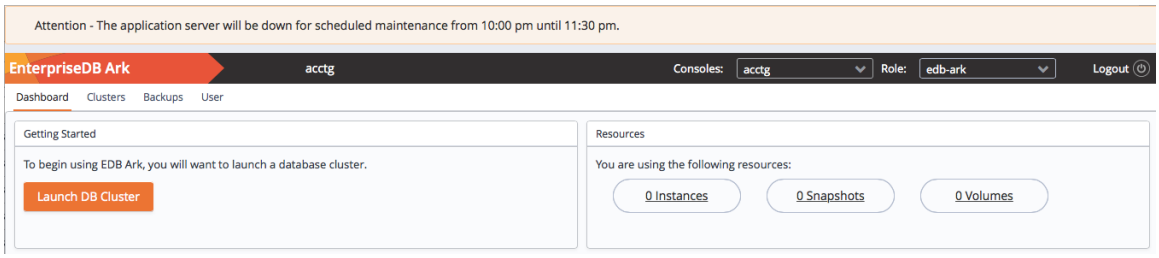

Figure 4.48 - Displaying a wall message.

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.

4.1.8 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 4.49).

Figure 4.49 – The Download Console Logs section of the Admin tab.

You can also review the console logs via an ssh session. Log files are stored in `/var/log/edb-ark`. The current log file is `/var/log/edb-ark/ark.log`.

You can use the Linux `tail` utility 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.

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

```
tail -F file_name
```

For more information about the console logs, please see Section 6.4.

4.1.9 Taking a Manual Backup of the Console

Use the `Backup Now` button in the `Backup Ark Console` panel of the `Admin` tab to request a manual backup of the Ark console begin (see Figure 4.50).

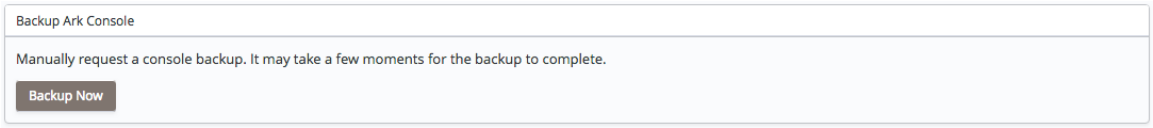

Figure 4.50 – The Backup Ark Console section of the Admin tab.

Click the `Backup Now` button to start a console backup; the backup will be uploaded to the currently configured object storage service.

4.1.10 Editing Installation Properties

Use the option displayed in the `Edit Installation Properties` section to review or modify Ark console properties (see Figure 4.51).

Figure 4.51 – The Edit Installation Properties section.

Click the `Edit installation properties` button to open the `Edit Installation Properties` dialog. Use fields on the `Edit Installation Properties` dialog to modify the properties of the Ark console. 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.

For detailed descriptions of each field:

- For an Amazon-hosted console, see Section 3.1.3.
- For an Azure-hosted console, see Section 3.2.5.

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 4.52) is accessible only to administrative users.

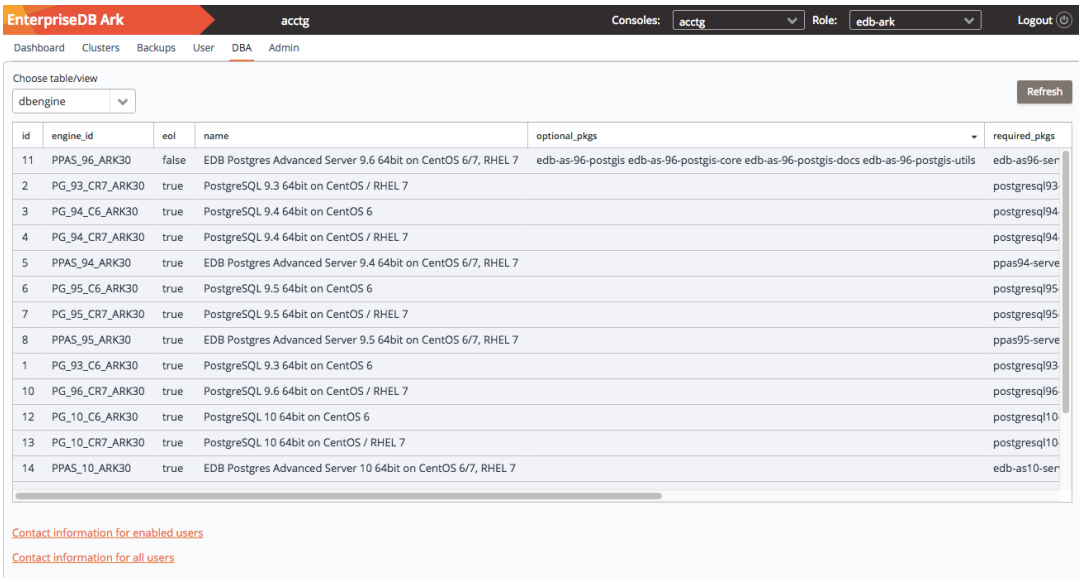

Figure 4.52 - The DBA tab.

Use the `Choose table/view` drop down listbox (shown in Figure 4.53) to select a view.

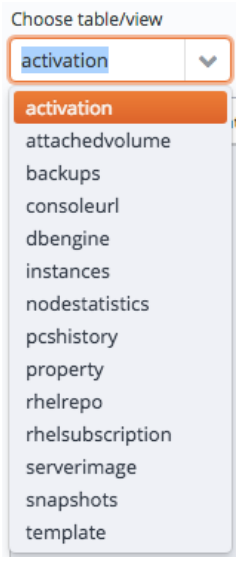

Figure 4.53 - The table/view listbox.

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

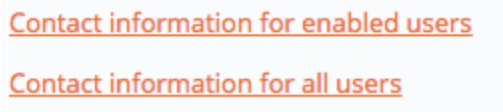

Figure 4.54 - The contact information links

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 status of the user account (`TRUE` if enabled, `FALSE` if disabled).
- 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 it is 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.

Column Name	Description
CREATIONTIME	The date and time that the node was created.
DATATHRESHOLD	The disk space threshold at which the cluster will be automatically scaled up.
DBNAME	The name of the default database created when the instance was created (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 the database that is running on the instance.
FREEDATASPACE	The current amount of free data space on the instance.
IMAGEID	The server image used when creating the node.
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.

Column Name	Description
ENCRYPTKEY	The encryption key for the cluster.
CLUSTERKEY	The SSH key shared by all of the instances in the cluster.
CLUSTERKEYNAME	The name of the SSH key.
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).

Column Name	Description
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.
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.

Column Name	Description
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.

4.3.14 template

The template table provides information about the configurations available with each template.

Column Name	Description
ID	The unique identifier of the template (system assigned).
BACKUPRETENTION	The number of backups that Ark will retain for the master node of the cluster.
BACKUPWINDOW	The time during which backups will be taken.
CONTINUOUSARCHIVING	True if archiving is enabled; false if archiving is disabled.
DESCRIPTION	The colon delimited namd of the cluster owner and the cluster.
DISABLED	True if the template is enabled; false if it is disabled.
ENCRYPTFS	True if the content of a backup is stored on an encrypted file system; false if it is not.
HARDWARE	The specified hardware size of the instance.
NAME	The name of the template.
NUMNODES	The number of nodes that will be created in a cluster deployed with this template.
STORAGE	The amount of disk space available to the cluster.
TENANTS	The names of the roles or tenants that may use this template.
VPC	The VPC ID (valid for AWS users only).
YUMUPDATE	True if updates are enabled for the cluster; false if they are not.
DBENGINE_ID	The database engine used by the template.
MANUALLYSCALEREPLICAS	MANUALLYSCALEREPLICAS indicates if users of this template will be allowed to manually scale replica nodes configured by this template.
MANUALLYSCALESTORAGE	MANUALLYSCALESTORAGE indicates if users of this template will be allowed to manually scale storage on clusters configured by this template.
AUTOSCALEREPLICAS	AUTOSCALEREPLICAS indicates if users of this template will be able to configure automatic node scaling for clusters configured by this template.
AUTOSCALESTORAGE	AUTOSCALESTORAGE indicates if users of this template will be able to configure automatic storage scaling for clusters configured by this template.
OPTIMIZED	Boolean value; true if an instance is optimized; false if not (valid for AWS users only).
IOPS	The IOPS setting for the cluster (valid for AWS users only).

5 Securing EDB Ark

Each cluster has an associated 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 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).

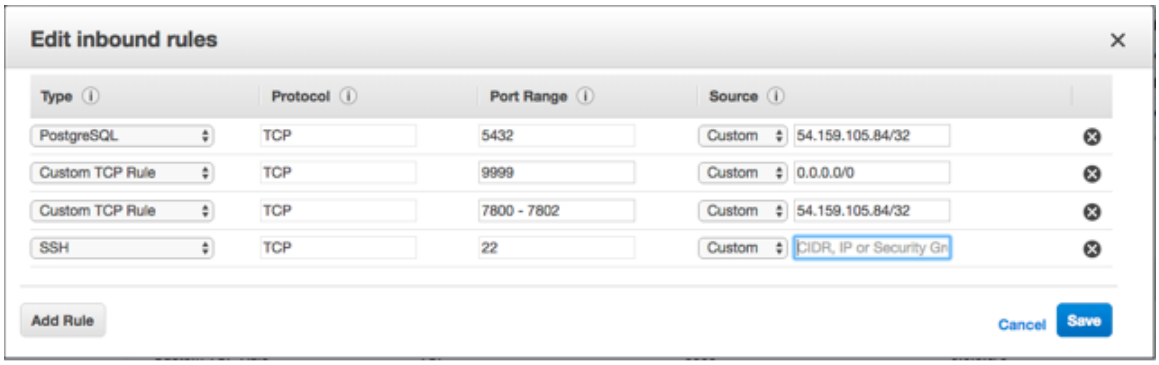

Figure 5.3 – Opening a port for an SSH connection.

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/10/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

5.2 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).

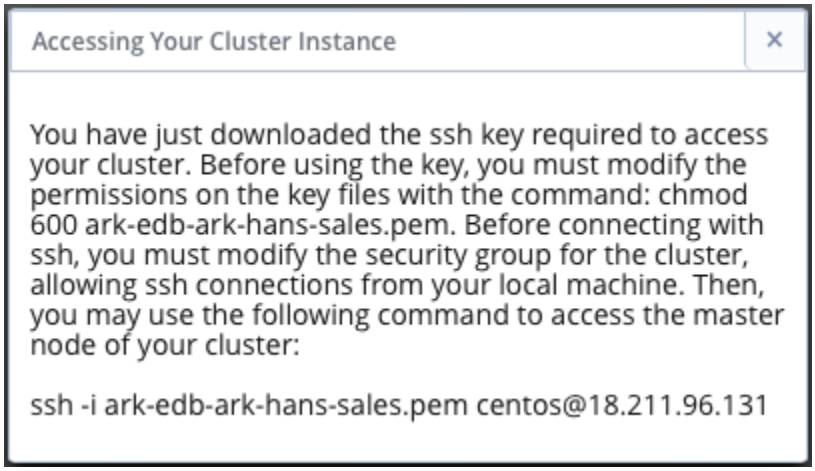

Figure 5.4 – Accessing Your Cluster Instance.

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
```

5.3 Using iptables Rules

If you are using iptables rules to manage security 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 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.

5.4 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 Ark Console

Apache Tomcat is an opensource project that deploys Java servlets on behalf of the Ark console. To start, stop, or restart the application server, use `ssh` to connect to the host of the Ark console database. Then, use `sudo` to assume sufficient privileges to restart the console:

```
sudo su -
```

Then, use `systemctl` to start, stop, or restart the server.

To start the server:

```
systemctl start tomcat
```

To stop the server:

```
systemctl stop tomcat
```

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

```
systemctl restart tomcat
```

6.2 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.

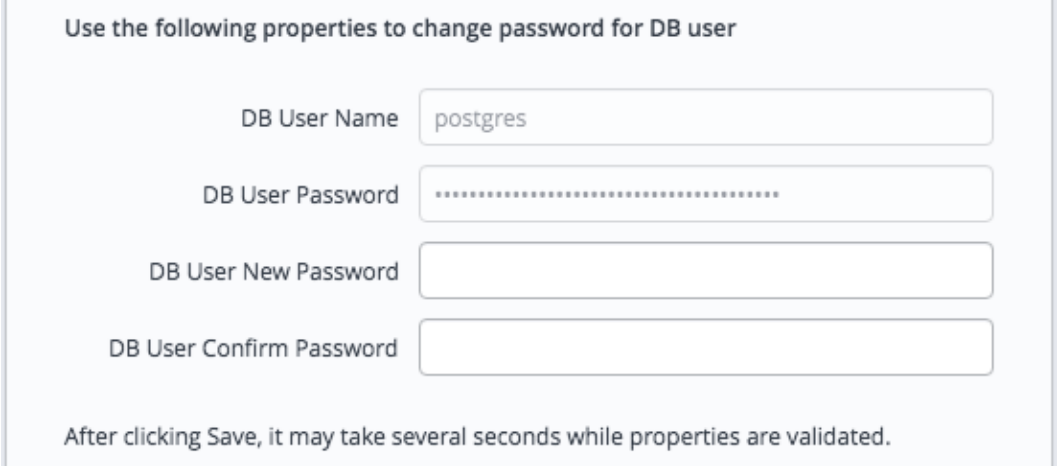

Figure 6.1 – Modifying the database password.

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.

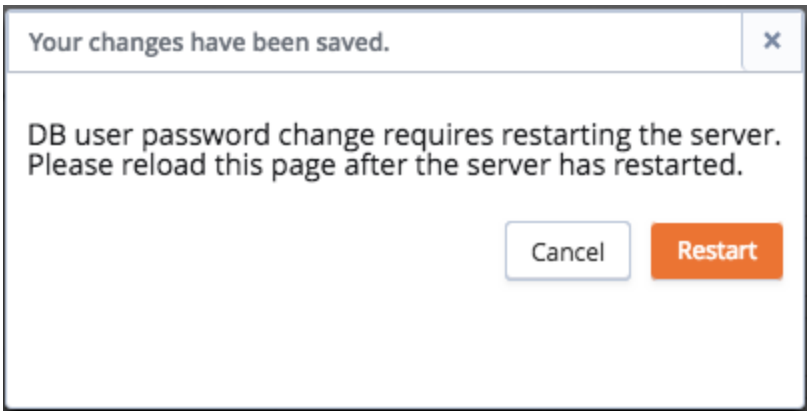

Figure 6.2 – Modifying the database password.

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

Please note: if you modify the password of the Ark console, the password of the PEM server that resides on the Ark console will change as well. When using the PEM web interface to connect to the PEM server, use the password assigned to the Ark console.

6.3 Customizing the Console

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

- Background images
- Background colors
- Fonts
- Font colors

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

Modifying the Console Display

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

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

The css File

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

```
/usr/share/tomcat/webapps/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:

```
/usr/share/tomcat/webapps/PPCDConsole/VAADIN/themes/pcsconsole/im
ages
```

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:

```
/usr/share/tomcat/webapps/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:

```
/usr/share/tomcat/webapps/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.

6.4 Managing Console Logs

By default, Ark console log files are written to `/var/log/edb-ark/ark.log`. Log files are rotated on a daily basis, and stored for 30 days.

You can use the `ark.server.level` property to manage the level of detail saved in the Ark console log files. The `ark.server.level` property resides in:

```
/usr/share/tomcat/webapps/PPCDConsole/WEB-INF/classes
```

To modify the value, connect to the Ark console, and use your choice of editor to modify the property value. The valid values are:

Property Value	Information Logged
SEVERE	Includes the least amount of information in the log files (i.e., exceptions and ERROR messages).
WARNING	Includes WARNING messages.
INFO	Includes informational messages about server activity.
CONFIG	Includes messages about configuration changes.
FINE	This is the default; provides detailed information about server activity.
FINER	Includes a higher level of detail about server activity.
FINEST	Provides the highest level of detail about server activity.

After modifying the properties file, restart the server to make the changes take effect:

```
sudo systemctl restart tomcat
```

6.5 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.

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 `/var/log/edb-ark/ark.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`), enabling 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 `/usr/share/tomcat/` directory:

   ```
   cd /usr/share/tomcat
   ```

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
updating changes for v3.2 on the console

--2018-07-25 16:15:34--
https://jdbc.postgresql.org/download/postgresql-42.2.4.jar
Resolving jdbc.postgresql.org (jdbc.postgresql.org)...
174.143.35.228, 2001:4800:1501:1::228

Connecting to jdbc.postgresql.org
(jdbc.postgresql.org)|174.143.35.228|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 814992 (796K) [application/java-archive]
Saving to: 'postgresql-42.2.4.jar'
100%[====>] 814,992     1.23MB/s   in 0.6s

2018-07-25 16:15:35 (1.23 MB/s) - 'postgresql-42.2.4.jar'
saved [814992/814992]

Script will upgrade the application! Is the EDB-ARK console
in a steady state (no logged in users, no activity in the
console)?
```

When prompted, enter `y` to perform the console upgrade.

```
Are you sure you want to continue? <y/N> y
Updating EDB-ARK Application...
Stopping httpd and tomcat services...
Deploying the latest application in tomcat
Starting httpd and tomcat services...

Done!
```

If the Ark upgrade 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`*. The package manager identifies any additional 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 will provide a list of any files that were in conflict.

6.6 Updating a PEM Installation on an Ark 3.0 Console

The PEM software initially distributed with Ark 3.0 is a development version. If you are using a local installation of PEM to monitor an Ark 3.0 console, you may want to update your version of PEM 7.3 to a more recent version. To update your PEM installation:

1. Ensure that the `edb-ark.repo` file is enabled and contains your connection credentials:

   Use ssh to connect to the Ark console host, and navigate to the `/etc/yum.repos.d` directory. Then, use your choice of editor to update the `edb-ark.repo` file.

   ```
   [edb-ark]
   name=EnterpriseDB EDB-ARK $releasever - $basearch
   baseurl=https://<username>:<password>@yum.enterprisedb.com/edb-
   ark/redhat/rhel-$releasever-$basearch
   enabled=0
   gpgcheck=1
   gpgkey=file:///etc/pki/rpm-gpg/ENTERPRISEDB-GPG-KEY

   [edb-tools]
   name=EnterpriseDB Tools $releasever - $basearch
   baseurl=https://<username>:<password>@yum.enterprisedb.com/tools/
   redhat/rhel-$releasever-$basearch
   enabled=0
   gpgcheck=1
   gpgkey=file:///etc/pki/rpm-gpg/ENTERPRISEDB-GPG-KEY

   [edb-dependencies]
   name=EnterpriseDB Dependencies $releasever -
   $basearchbaseurl=https://<username>:<password>@yum.enterprisedb.c
   om/dependencies/redhat/rhel-$releasever-$basearch
   enabled=0
   gpgcheck=1
   gpgkey=file:///etc/pki/rpm-gpg/ENTERPRISEDB-GPG-KEY
   ```

   Modify the repository details, replacing each `<username>` and `<password>` placeholder with your credentials for the EnterpriseDB repository, and setting `enabled` to `1`.

   To request credentials for the repository, please visit:

   https://www.enterprisedb.com/repository-access-request

2. Assume superuser privileges.

   ```
   sudo su
   ```

3. Use yum to update the PEM agent and PEM server version installed on your Ark console host:

   ```
   yum update edb-pem edb-pem-server
   ```

4. When the installation completes, configure the updated PEM server:

   ```
   /usr/edb/pem/bin/configure-pem-server.sh -dbi /usr/pgsql-10
   -d /var/lib/pgsql/10/data/ -ho 127.0.0.1 -p 5432 -su
   postgres -sp ark_console_password -t 1 -ci 0.0.0.0/0 -ds
   postgresql-10 -acp ~/.pem/
   ```

   Where *ark_console_password* is the password of the Ark console.

   For more information about using PEM, please see the PEM user guides, available at:

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

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.

Please note: the Ark recovery utility only supports recovering the console from a backup that is the same version as the current console version.

7.1 Modifying Backup Properties with the EDB Ark Console

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

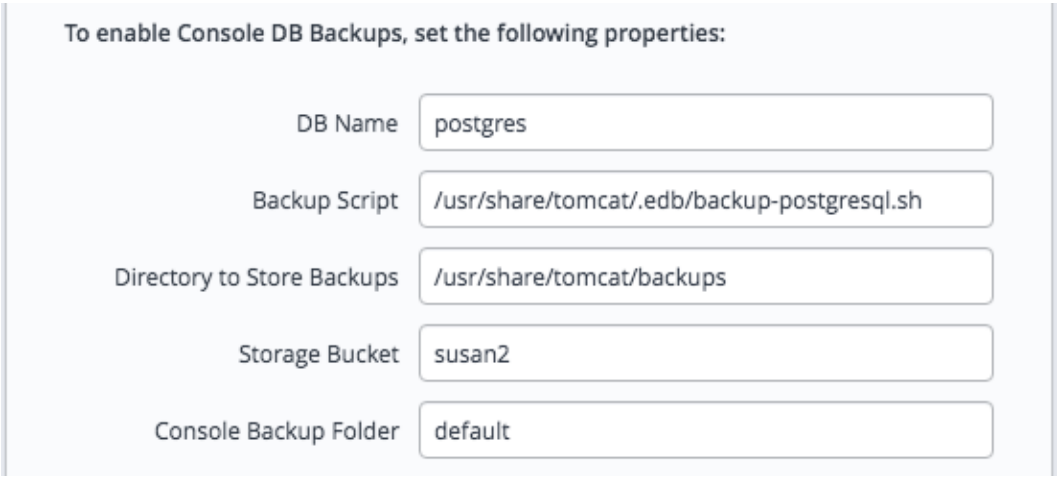

Figure 7.1 – The console backup properties.

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.

- 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 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

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).

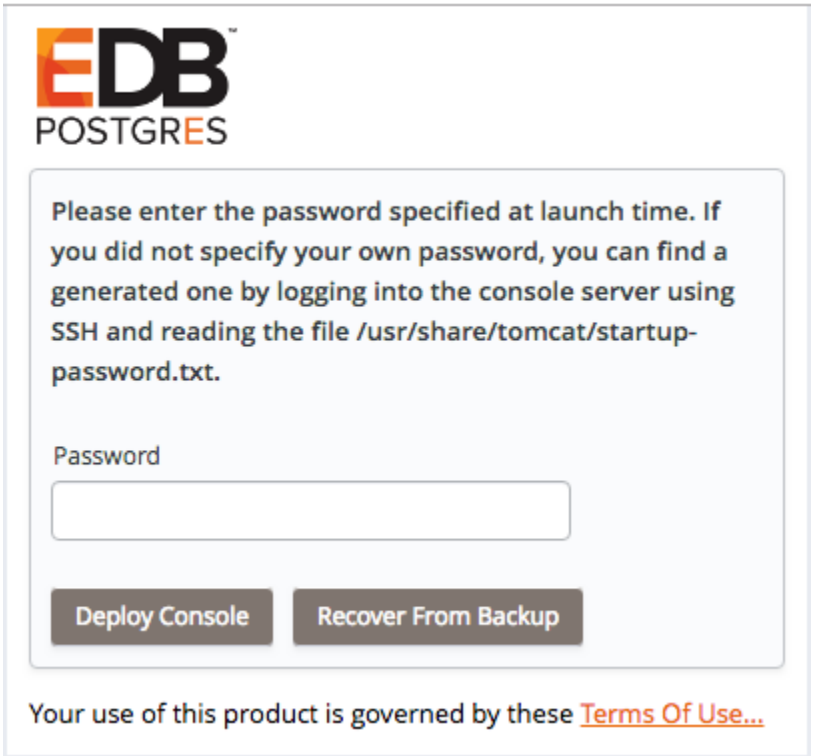

Figure 7.2 - The connection dialog.

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.

Use the dialog to provide details about the console, and the location of a backup to recover. 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.

Use the `Folder name` drop-down listbox to select the backup you wish to use for the recovery, and click `Finish` to start the recovery process. If your system is monitored by a local PEM server, the backup will attempt to restore both the Ark and PEM servers.

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	Data storage scaling for cluster *cluster_name* has been suspended.

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 Container Created	A storage container (bucket) named *bucket_name* has been created. All EDB Ark clusters configured for Continuous Archiving (Point-in-Time 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 the server.
Ark Synchronization Error With PEM Server	The Ark console has encountered an error while attempting to synchronize with the PEM server: *exception_information* Ark console: *https: console_address* PEM dashboard: https://*pem_console_address*/pem/browser/ Full Error Details: *details*
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 - Amazon AWS Policies

10.1 Reference - Amazon 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 Amazon IAM Role Trust Relationship

When you define an Amazon IAM 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"
        }
      }
    }
  ]
}
```

10.3 Reference – AWS IAM Role Permission Policy

When you define an Amazon user, 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:::*",
"Effect": "Allow",
"Sid": "Stmt1407961734627"
}, {
"Condition": {
"StringEquals": {
"ec2:ResourceTag/CreatedBy": "EnterpriseDB"
}
},
"Action": [
"ec2:RebootInstances",
"ec2:StopInstances",
"ec2:TerminateInstances"
],
"Resource": "*",
"Effect": "Allow",
"Sid": "Stmt1407961927870"
```

```
}
]
}
```

11 Creating a Statically Provisioned Image

An `install.sh` script is distributed with Ark; 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 10 database on a RHEL host:

```
$ sudo ./install.sh -n ppas -v 10 -u true -c true \
-r http://USERNAME:PASSWORD@yum.enterprisedb.com/10/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-as10-server edb-pgpool35 edb-as10-pgpool35-extensions" \
```

The following command creates a static image that contains PostgreSQL 10 on a CentOS host:

```
$ sudo ./install.sh -n postgres -v 10 -u true -c false \
-r http://yum.postgresql.org/10/redhat/rhel-7-x86_64/pgdg-redhat96-10-
3.noarch.rm \
-p "postgresql10-server pgpool-II-10" \
```

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 .