Ark Platform

Version 3.5

Built at 2021-07-20T10:22:43

1 Ark

EDB Postgres Ark

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.

1.1 What's New

The following features have been added to the EDB Ark user console for this release.

- Ark now supports EDB Postgres Advanced Server and PostgreSQL version 12 database clusters.
- You can use the `Promote` option (located on a cluster's context menu in the `Details` panel) to replace the master node with a standby node. For more information, see the EDB Postgres Ark Getting Started Guide.
- The `/templates` and `/templates/id` resources now support the `region` property.

Limitations

Cloning, Recovering, or Scaling Encrypted Clusters from Previous Versions

Encrypted clusters created with Ark 3.3 or prior may not be used to clone, recover, or scale to a machine type that is not supported by that earlier version. If you will be moving encrypted clusters created with Ark version 3.3 or prior to a new machine type (as supported by Ark 3.4), you will need to:

1. Clone the encrypted cluster to a new, unencrypted cluster.
2. Upgrade the Ark console on which the cluster resides to Ark 3.4.
3. Clone the unencrypted cluster to a new encrypted cluster of the new machine type.

For detailed information about cloning a cluster, see the *EDB Postgres Ark Getting Started Guide*.

Cloning with a Template from a Foreign Region

You cannot use a template when cloning from a foreign region.

Monitoring a Federated Console with PEM

Ark does not support federated consoles configured in local PEM server mode.

Federating Consoles that host clusters with the Same Name

If you are upgrading to version 3.4, and plan to federate existing Ark version 3.3 consoles that host clusters with the same name, you must first create a clone of one of the clusters specifying an alternate name; federated consoles cannot be used to host clusters with the same name. For example, if you plan to federate two consoles that both contain a cluster named `acctg`, you should clone one of the clusters specifying an alternate name for the cluster (i.e. `acctg-west`). Then, you can federate the consoles; both consoles will have access to `acctg` and `acctg-west`.

1.2 Supported Platforms

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

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

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

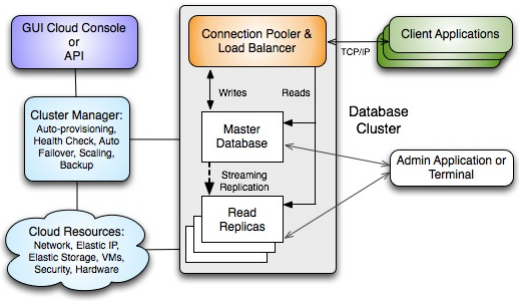
- RHEL 7.x
- CentOS 7.x

1.3 Ark Architecture Overview

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

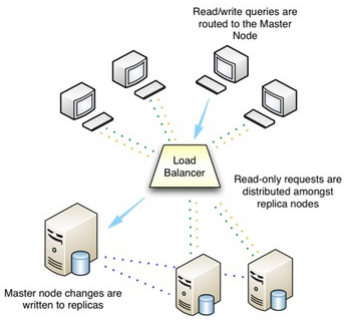
!!! Note Traditionally, the expression *cluster* refers to a single instance of Postgres managing multiple databases; an EDB Ark database server cluster is a collection of high-availability Postgres server instances that reside in a cloud or on a traditional network.

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.



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. 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.



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.

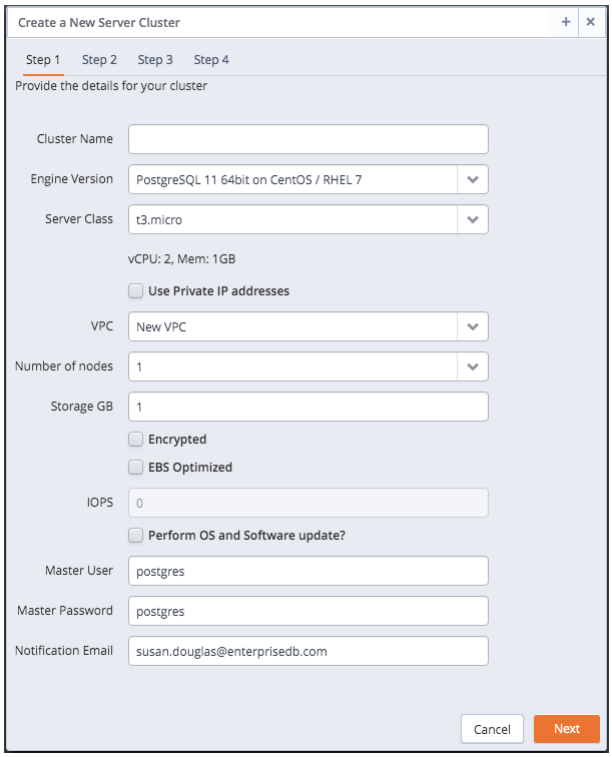
EDB Ark makes it easy to scale a database cluster:

- 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.

Using Ark on an 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.



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 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.

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.

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

1.4 Registering an Ark Cluster with PEM

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.

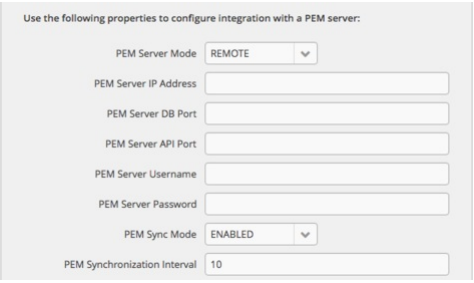


Fig. 4.1: Specifying Deployment Details

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.
  - Provide the IP address of the PEM server host in the `PEM Server IP 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.

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 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.

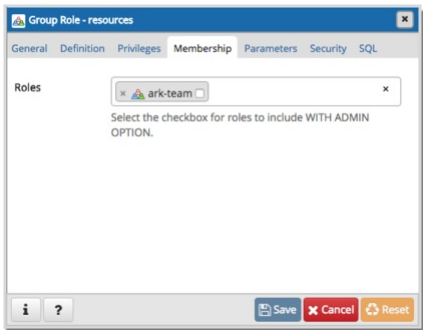


Fig. 4.2: The PEM Group Role dialog

- 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.

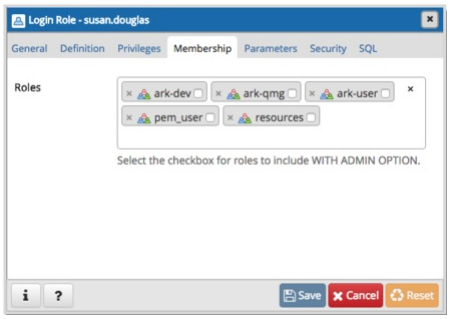


Fig. 4.3: The PEM Login Role dialog

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.

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.

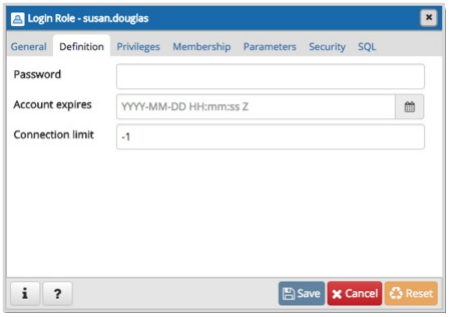


Fig. 4.4: Registering as a login role

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.

Monitoring an Ark Cluster

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, provide the PEM credentials to connect.

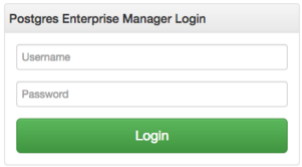


Fig. 4.5: Logging on to the PEM Server

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).

After authenticating with the server, the PEM web interface allows you to manage your monitored nodes.

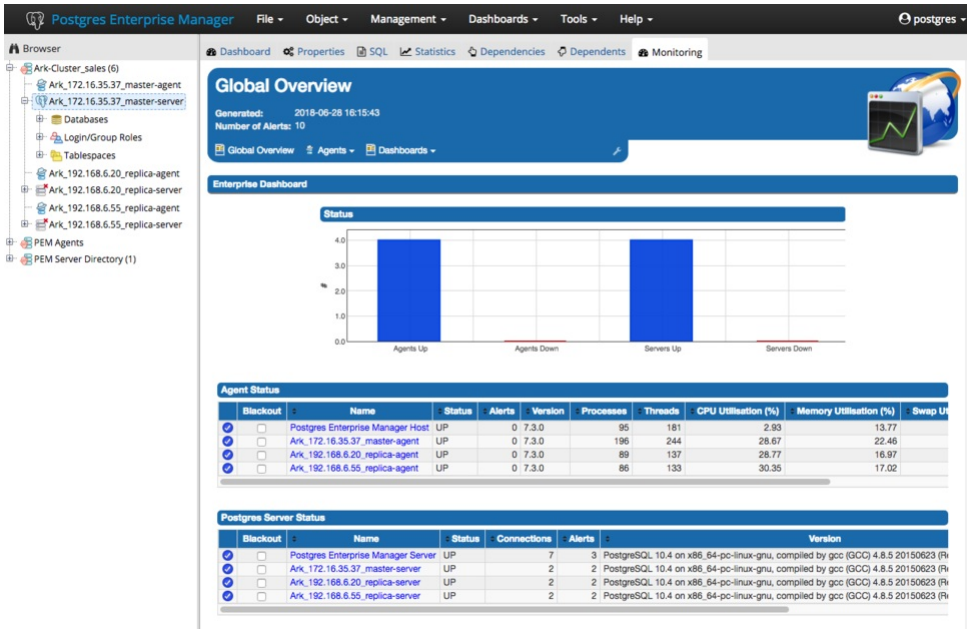


Fig. 4.6: The PEM Global Overview

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.

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

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 PEM 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.

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.

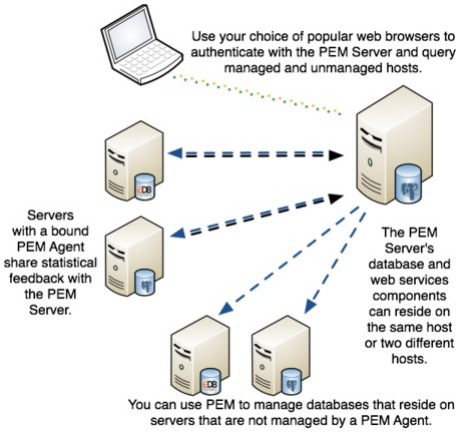


Fig. 4.7: PEM Architecture

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 installing and configuring PEM, please visit the EnterpriseDB website at www.enterprisedb.com.

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.

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 Ark console's `Dashboard` tab.

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 of the PostgreSQL database that stores PEM statistics to allow connections from any monitored servers as well as the PEM client.

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.

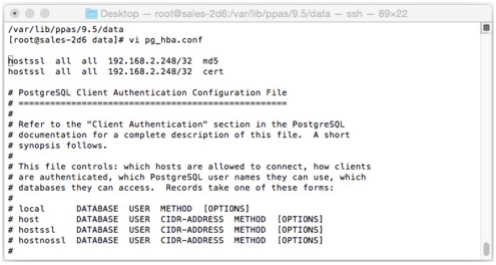


Fig. 4.8: Modifying the pg_hba.conf file

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, the following command restarts a PostgreSQL 10 service:

```
service postgresql-10 restart
```

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

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.

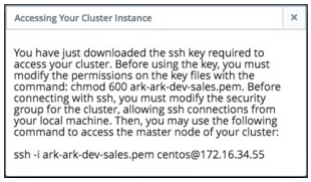


Fig. 4.9: Connecting with SSH

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.

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/<x.x>/data`, where `<x.x>` specifies the Ark server version.

Add entries to the `pg_hba.conf` file that allow connections from the PEM server.

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.

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.

Fig. 4.10: The agent is registered successfully

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 .

1.5 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 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 documentation available at the EnterpriseDB website.

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

- On 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. 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 documentation.

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.

Using Provider Authentication on AWS

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.

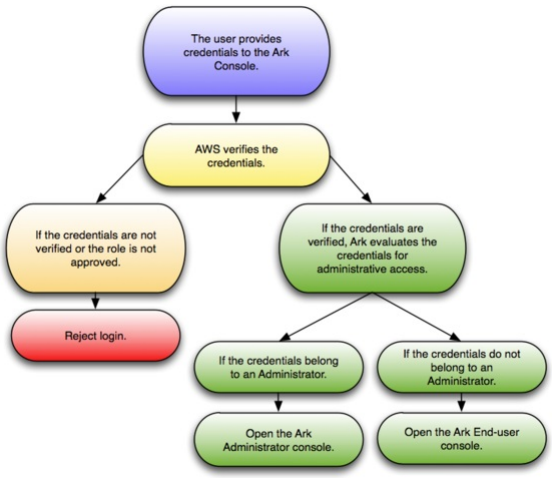


Fig. 5.1: Using Provider Authentication on AWS

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.

Using PostgreSQL Authentication on AWS

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

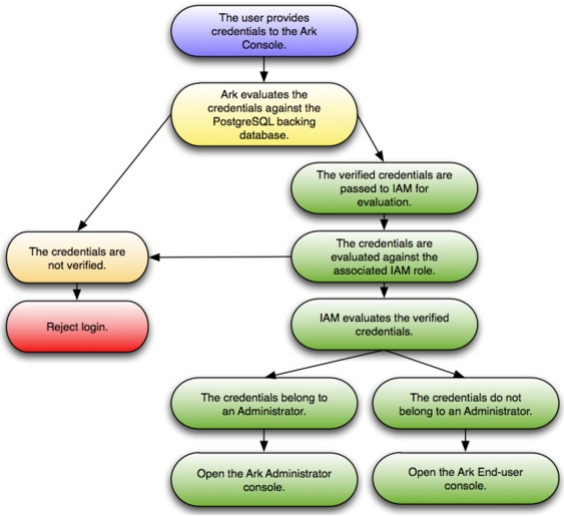


Fig. 5.2: Using PostgreSQL Authentication on AWS

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

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.

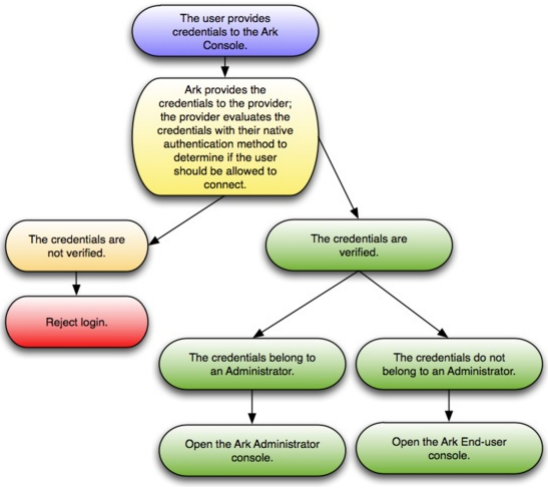


Fig. 5.3: Using Provider Authentication on Azure

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.

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.

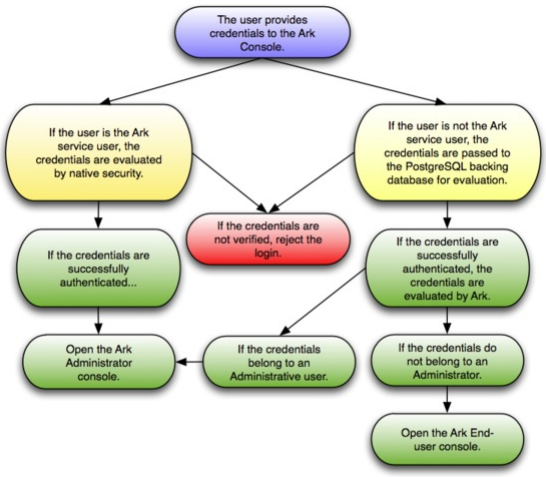


Fig. 5.4: Using PostgreSQL Authentication on Azur

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.

1.6 Installing the 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 to deploy an Ark console:

- If your cluster resides on an Amazon public cloud, see Installing the Ark Console on AWS for more information.
- If your cluster uses an Azure host, see Installing the Ark Console on Azure for more information.

1.6.1 Installing the Ark Console on 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:

- Launch an Ark instance with an Amazon AWS Marketplace AMI.
- Create an Amazon role and register an administrative user .
- Configure the Ark console .
- Create an Amazon role and register an Ark console user .

Launching the Ark Console Instance on AWS

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.

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.

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

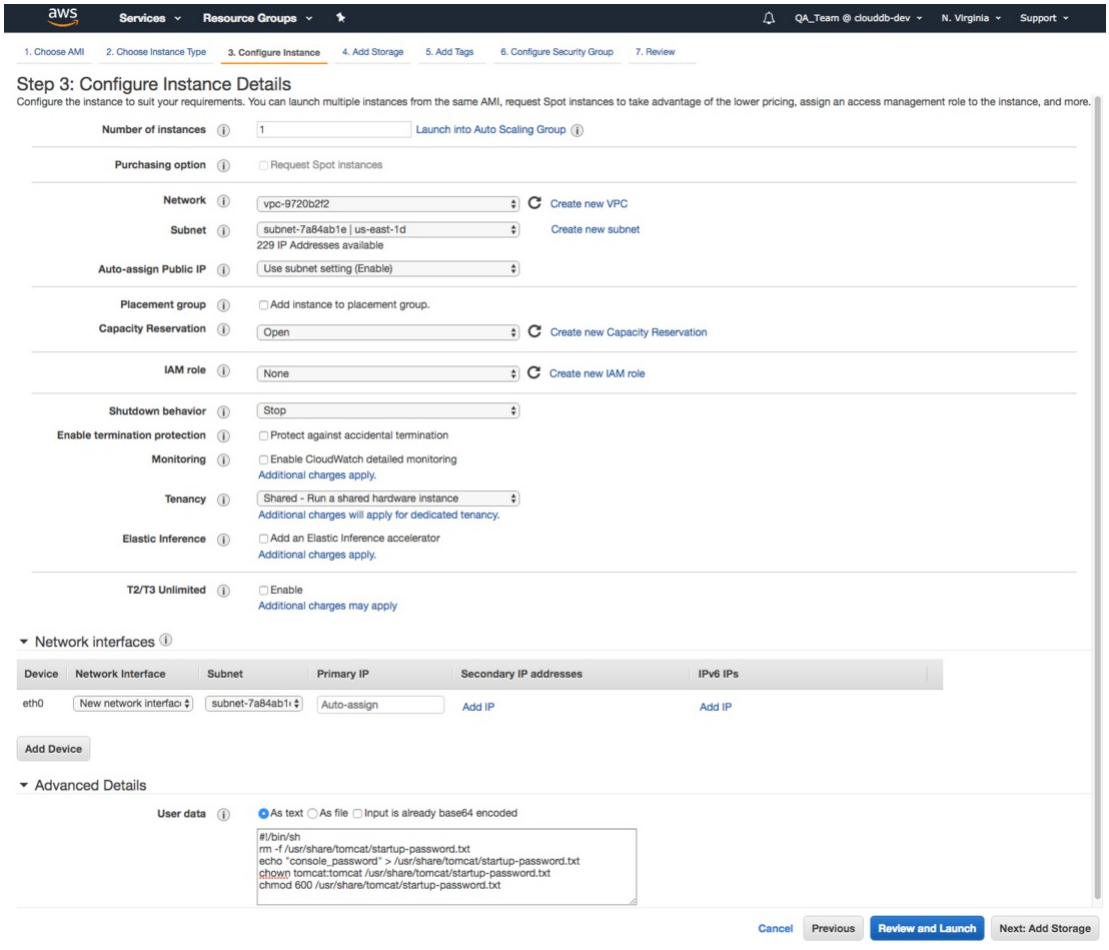


Fig. 6.1: Step 3 - Enabling the startup script

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

- 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
```

Replace the `<console_password>` variable in the script with a password for the console. 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 in the table below.

Rule Type	Direction	Port or Range	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	7800 to 7999	CIDR	0.0.0.0/0
Custom TCP	Ingress	5432	CIDR	0.0.0.0/0

Please Note:

- 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.

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.

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.

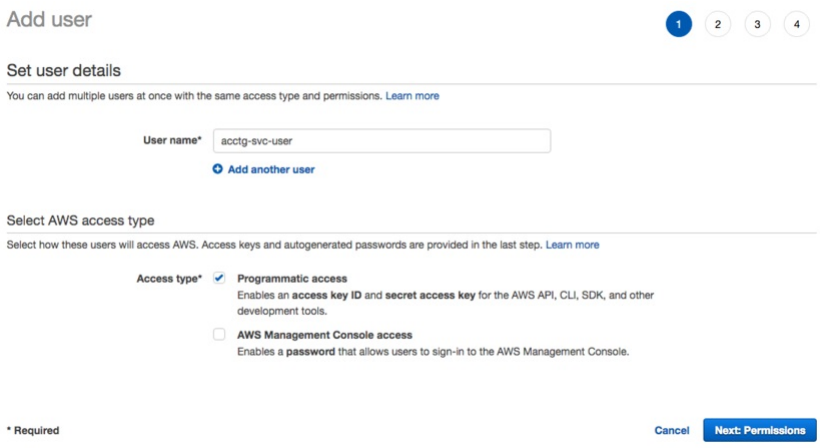


Fig. 6.2: The AWS Add user dialog

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.

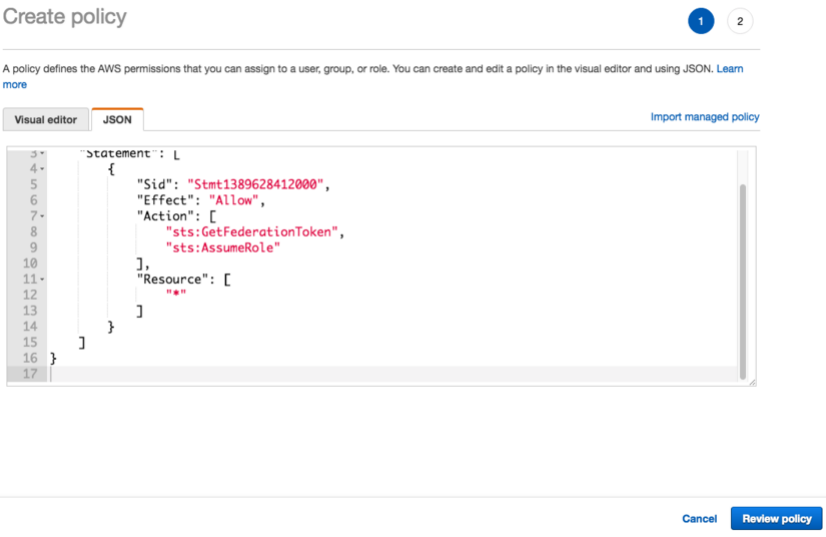


Fig. 6.3: Adding a policy definition

After copying in a policy, click the `Review policy` button to continue.

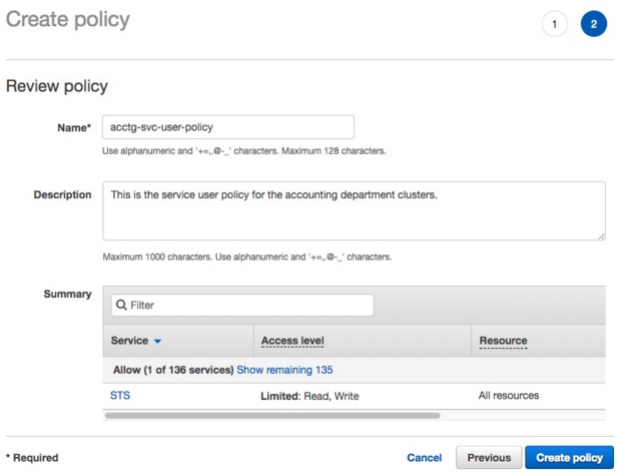


Fig. 6.4: Completing the policy definition

Provide a name and a description for the service policy definition, and click the `Create policy` button to continue.

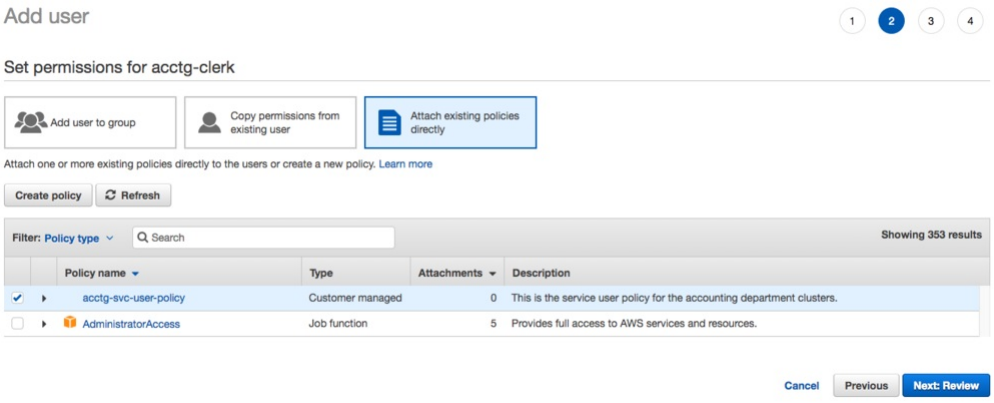


Fig. 6.5: Attaching the policy

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`.

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

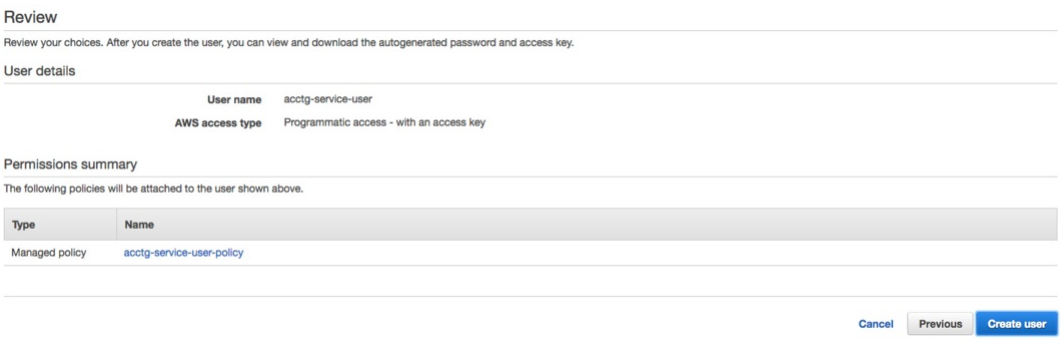


Fig. 6.6: Creating the user

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

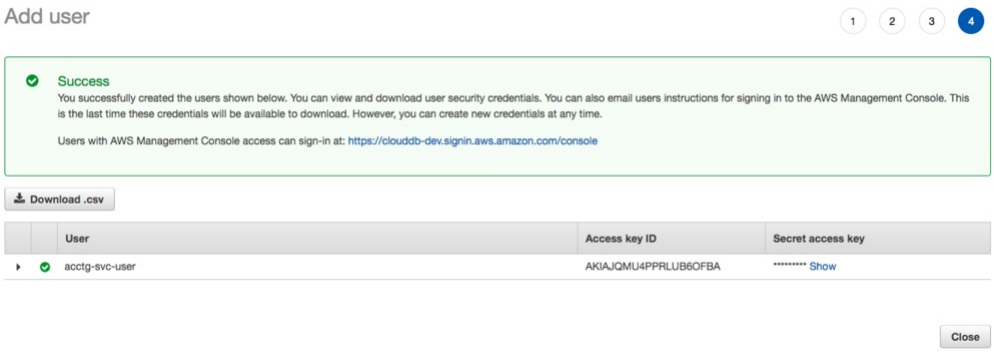


Fig. 6.7: The user is created successfully

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.

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.

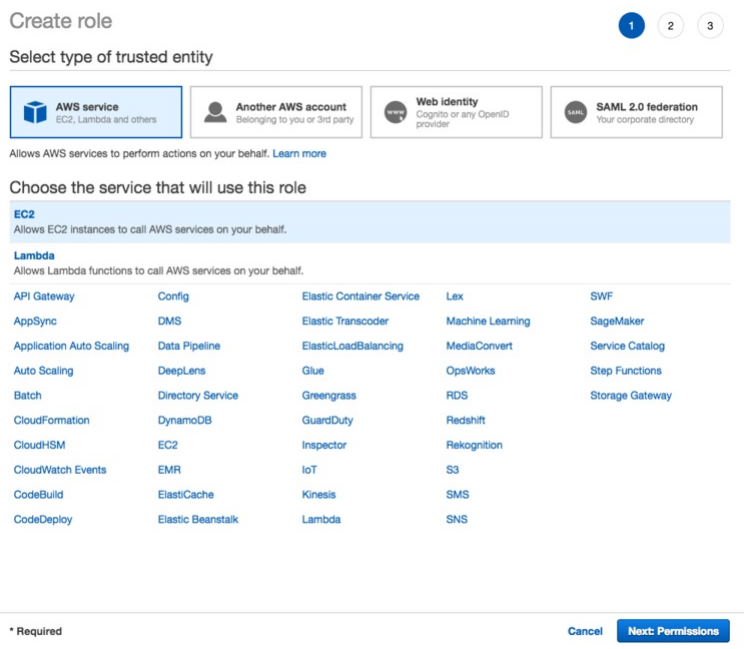


Fig. 6.8: Creating a role

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

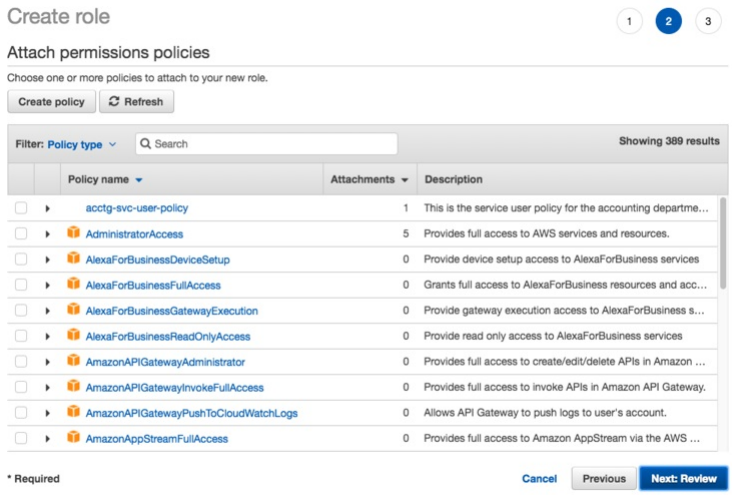


Fig. 6.9: The AWS Attach permissions policies dialog

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

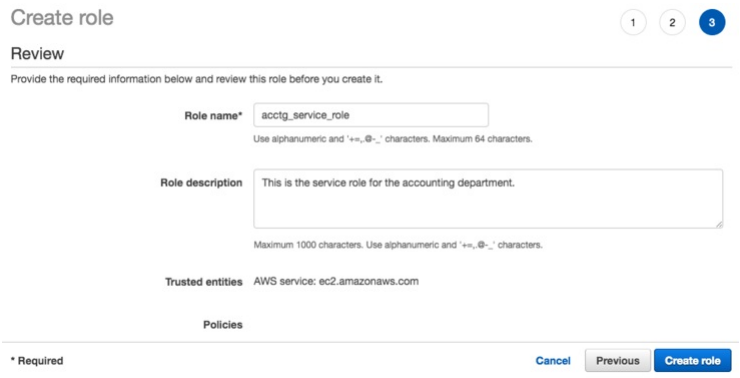


Fig. 6.10: Provide a role name and description

When the `Review` dialog opens, 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.

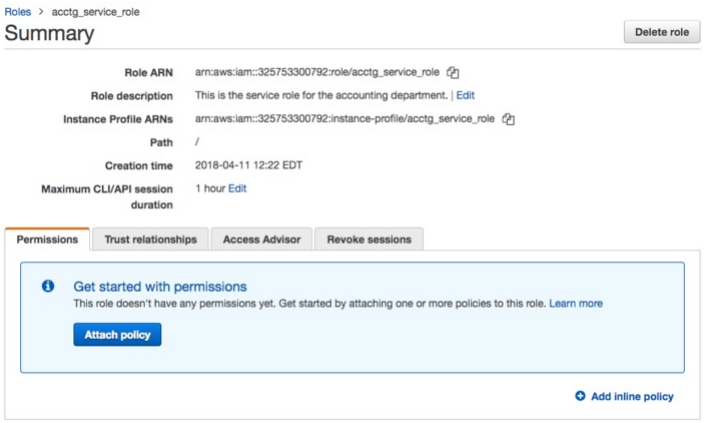


Fig. 6.11: The AWS Summary dialog

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.

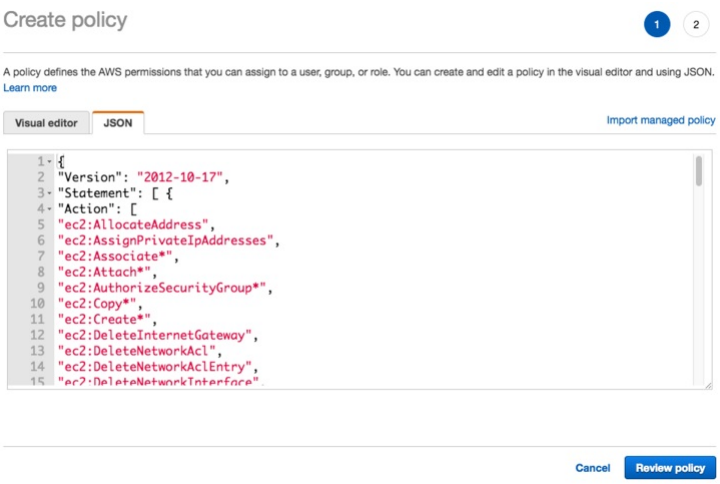


Fig. 6.12: Provide the policy name and contents

Copy the provided security policy into the `JSON` tab on the `Create policy` dialog. After providing security policy information, click `Review Policy` to provide a name for the policy, and return to the role information page.

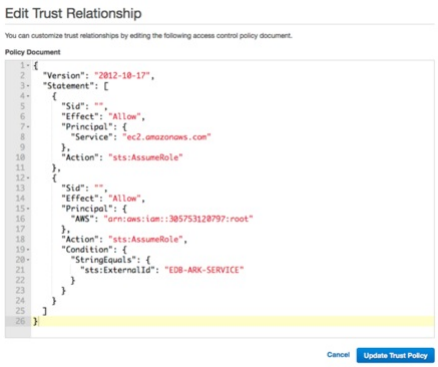


Fig. 6.13: The policy document

Navigate to the `Trust relationships` tab, and select the `Edit Trust Relationship` button to display the `Policy Document`. Replace the displayed content of the policy document with the provided Amazon IAM Role Trust Relationship.

Click the `Update Trust Policy` button to finish.

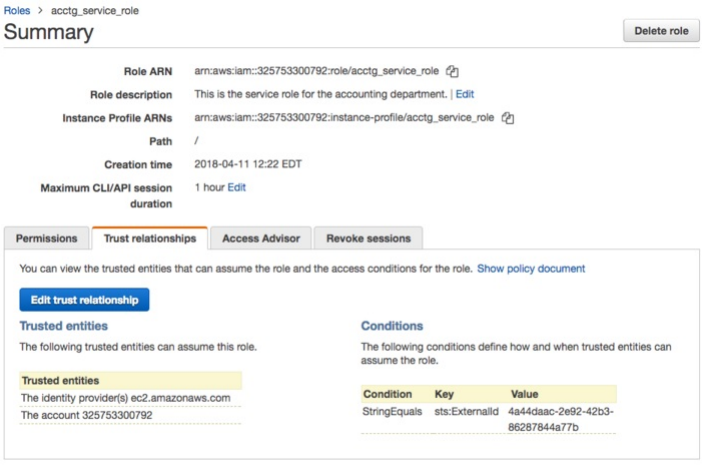


Fig. 6.14: The role detail summary

The role detail `Summary` 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.

Configuring the Ark Console

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

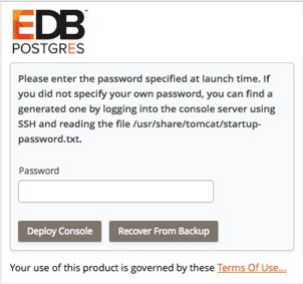


Fig. 6.15: Deploying the console

When prompted, provide the password specified when launching the console, and click `Deploy Console`.

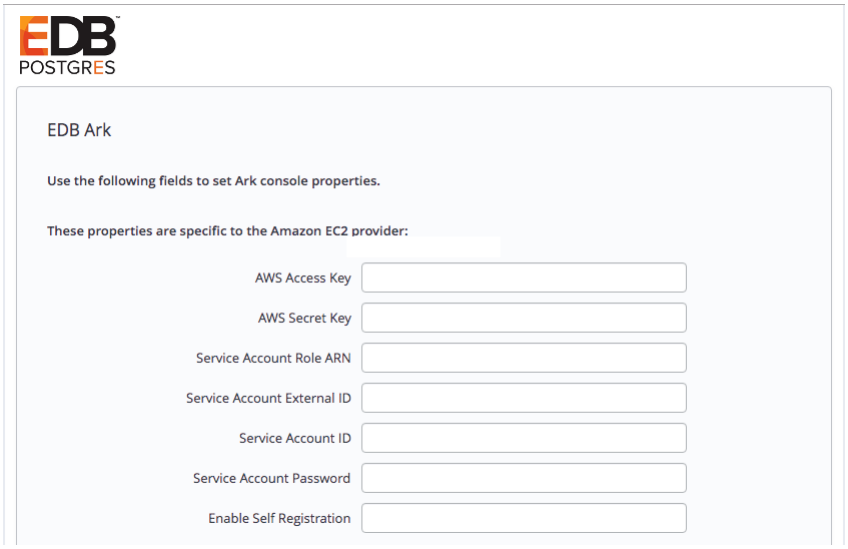


Fig. 6.16: The provider specific console properties

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

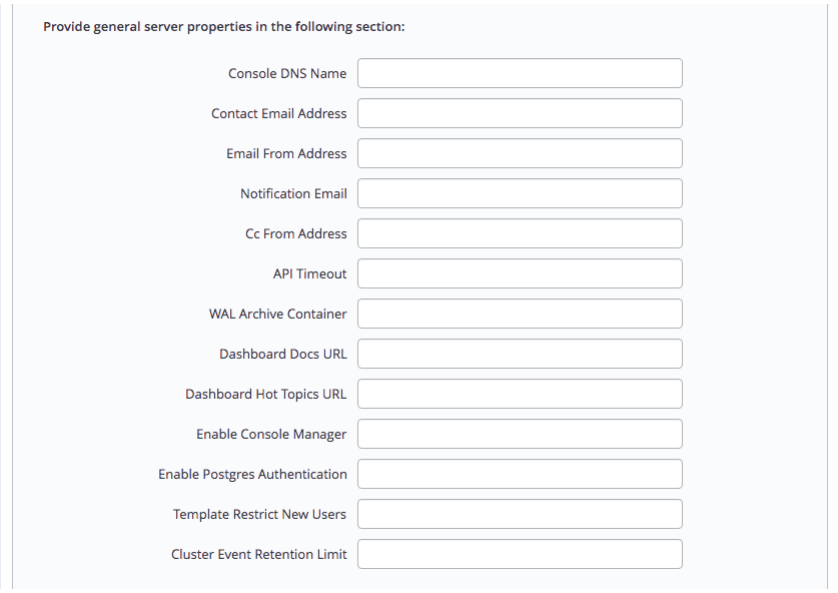


Fig. 6.17: Provide general server properties

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.

- 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.
- Use the `Cluster Event Retention Limit` field to specify how long the console will keep events for deleted clusters. The default value is 14 days.

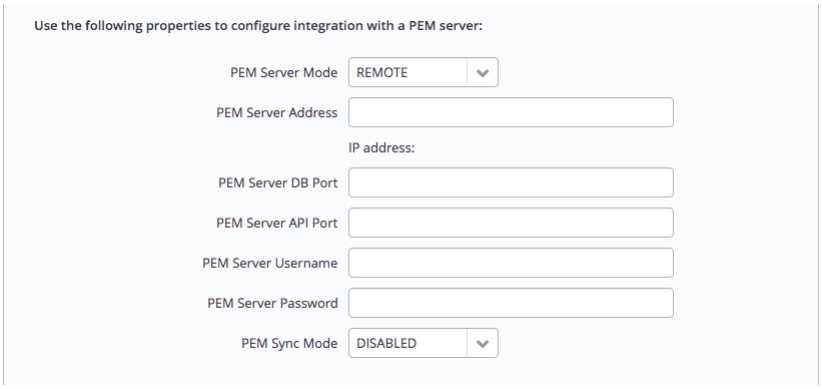


Fig. 6.18: Provide connection details for PEM

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 IP address of the PEM server host in the `PEM Server IP 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.
- Use the `PEM Synchronization Interval` field to specify the number of minutes between attempts to synchronize the Ark console with the PEM server.

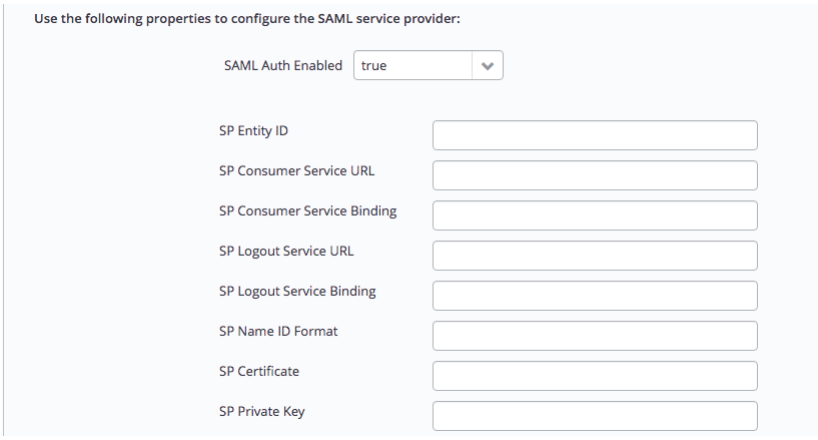


Fig. 6.19: Provide information about the service provider

If you specify `true` in the `SAML auth enabled` field, the Ark console will display the properties required to use SAML authentication when connecting to the Ark console. Use fields on the deployment dialog to specify SAML authentication properties. Use fields in the next section to provide information about the service provider:

- Use the `SP Entity ID` field to provide a URI that specifies the identifier of the service provider.
- Use the `SP Consumer Service URL` field to specify the URL from which the response from the identity provider will be returned.
- Use the `SP Consumer Service Binding` field to specify the SAML protocol binding to be used when returning the response message from the identity provider.
- Use the `SP Logout Service URL` field to specify the URL to which the service provider will specify information about where and how the `Logout Response` message MUST be returned to the requester, in this case our service provider.
- Use the `SP Logout Service Binding` field to specify the SAML protocol binding to be used when returning the `LogoutResponse` or sending the `LogoutRequest` message.
- Use the `SP Name ID Format` field to specify the constraints on the name identifier that will be used to represent the requested subject.
- Use the `SP Certificate` field to specify certificate information; this is usually `x509cert`. The private key of the service files are provided by files placed in the `certs` folder.
- Use the `SP Private Key` field to specify the location of the service providers private key; this must be in `PKCS#8` format.

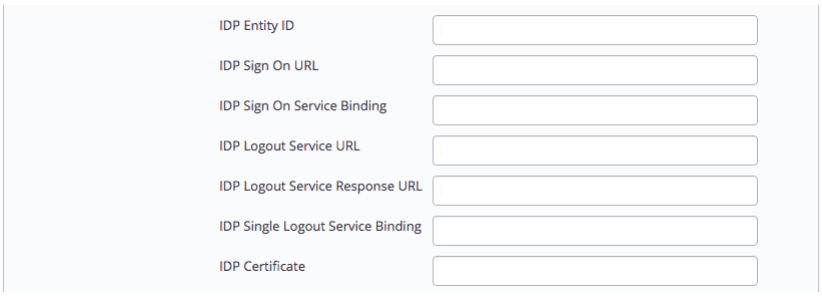


Fig. 6.20: Provide information about the identity provider

Use fields in the next section to provide information about the identity provider:

- Use the `IDP Entity ID` field to specify the identifier of the identity provider (this must be a URI).
- Use the `IDP Sign On URL` field to specify the URL target of the identity provider (where the service provider will send the Authentication Request Message).
- Use the `IDP Sign On Service Binding` field to specify the SAML protocol binding to be used when returning the response message. This version of Ark only supports the HTTP-Redirect binding.
- Use the `IDP Logout Service URL` field to specify the URL Location of the identity provider to which the service

provider will send a single logout Request.

- Use the `IDP Logout Service Response URL` field to specify the URL Location of the identity provider to which the service provider will send a single logout response.
- Use the `IDP Single Logout Service Binding` field to specify the SAML protocol binding to be used when returning the response message.
- Use the `IDP Certificate` field to specify the Public x509 certificate of the identity provider.

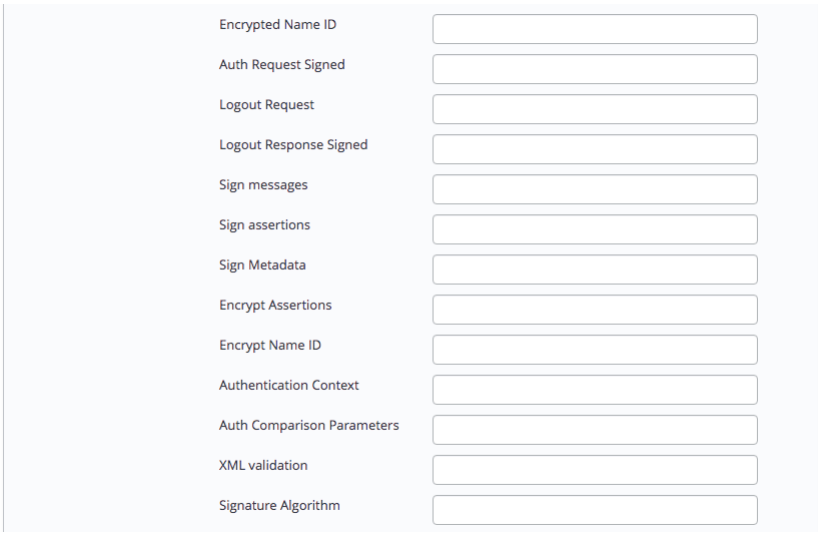


Fig. 6.21: Provide information about your SAML preferences

Use fields in the next section to provide your SAML preferences:

- Use the `Encrypted Name ID` field to indicate that the name identifier of the `samlp:logoutRequest` sent by this service provider will be encrypted; specify `true` or `false`.
- Use the `Auth Request Signed` field to indicate if the `samlp:AuthnRequest` messages sent by this service provider will be signed; specify `true` or `false`.
- Use the `Logout Request` field to indicate if the `samlp:logoutRequest` messages sent by the service provider will be signed; specify `true` or `false`.
- Use the `Logout Response Signed` field to indicate if the `samlp:logoutResponse` messages sent by the service provider will be signed; specify `true` or `false`.
- Use the `Sign messages` field to sign the metadata. If you leave the field empty, the metadata will not be signed. If you wish to provide a signature, provide a comma separated `keyFileName`, `certFileName` pair.
- Use the `Sign assertions` field to indicate a requirement for the `samlp:Response`, `samlp:LogoutRequest`, and `samlp:LogoutResponse` elements received by the service provider to be signed; specify `true` or `false`.
- Use the `Sign Metadata` field to indicate that the metadata of this service provider must be signed; specify `true` or `false`.
- Use the `Encrypt Assertions` field to indicate that the assertions received by this service provider must be encrypted; specify `true` or `false`.
- Use the `Encrypt Name ID` field to indicate that the name identifier received by this service provider must be encrypted; specify `true` or `false`.
- Use the `Authentication Context` field to specify that `Set Empty` and `no AuthContext` will be sent in the `AuthNRequest`. You can set multiple values in a comma-delimited list.
- Use the `Auth Comparison Parameters` field to specify that the `authn` comparison parameter to be set; this field defaults to `exact`.
- Use the `XML validation` field to indicate if the service provider will validate all received xmls; specify `true` or `false`.
- Use the `Signature Algorithm` field to specify the algorithm that the toolkit will use for the signing process. Specify one of the following:
  - `http://www.w3.org/2000/09/xmldsig#rsa-sha1`

- `http://www.w3.org/2000/09/xmldsig#dsa-sha1`
- `http://www.w3.org/2001/04/xmldsig-more#rsa-sha256`
- `http://www.w3.org/2001/04/xmldsig-more#rsa-sha384`
- `http://www.w3.org/2001/04/xmldsig-more#rsa-sha512`

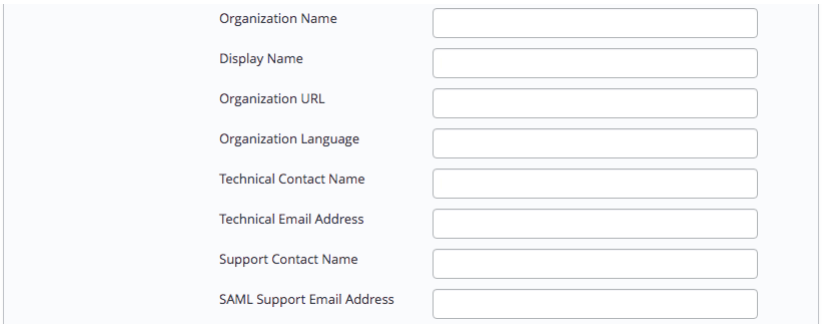


Fig. 6.22: Provide information about your organization

Use fields in the next section to provide information about your organization:

- Use the `Organization Name` field to specify the name of the organization for which authentication is being provided.
- Use the `Display Name` field to specify the display name of the organization.
- Use the `Organization URL` field to specify the URL of the organization.
- Use the `Organization Language` field to specify the primary language used by the organization.
- Use the `Technical Contact Name` field to specify the name of a technical contact.
- Use the `Technical Email Address` field to specify a contact email address for the technical contact.
- Use the `Support Contact Name` field to specify the name of a support contact.
- Use the `SAML Support Email Address` field to specify the email address of the SAML support contact.

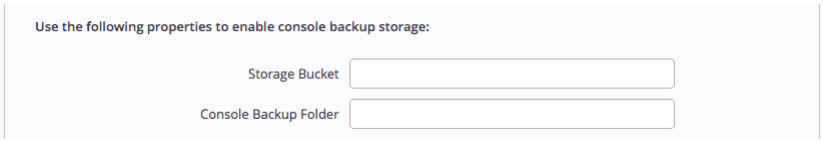


Fig. 6.23: Provide console backup information

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.

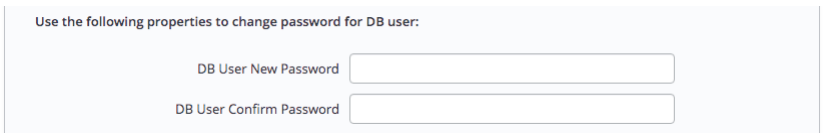


Fig. 6.24: Specify console password preferences

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`).

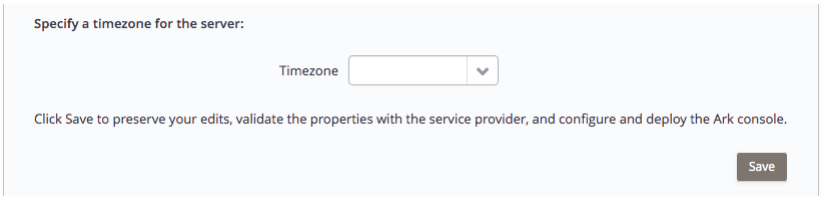


Fig. 6.25: Select a timezone for the server

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.

Creating an Amazon Role and Registering an Ark Console User

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

To define an Amazon role, connect to the Amazon management console, and navigate through the `Identity and Access Management` dashboard to the `Roles` dashboard, and click the `Create role` button.

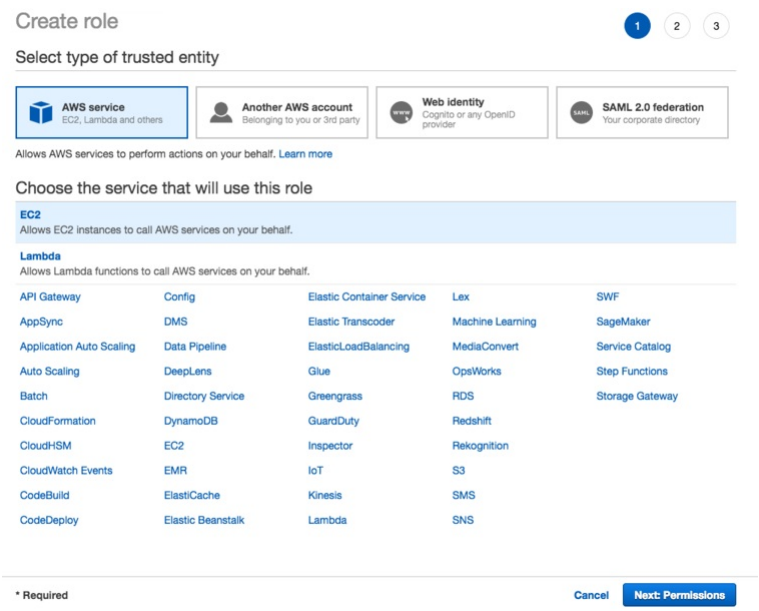


Fig. 6.26: The Create role dialog

When the `Create role` dialog opens, select the `AWS service button` and highlight the `EC2` bar, and click `Next: Permissions` to continue.

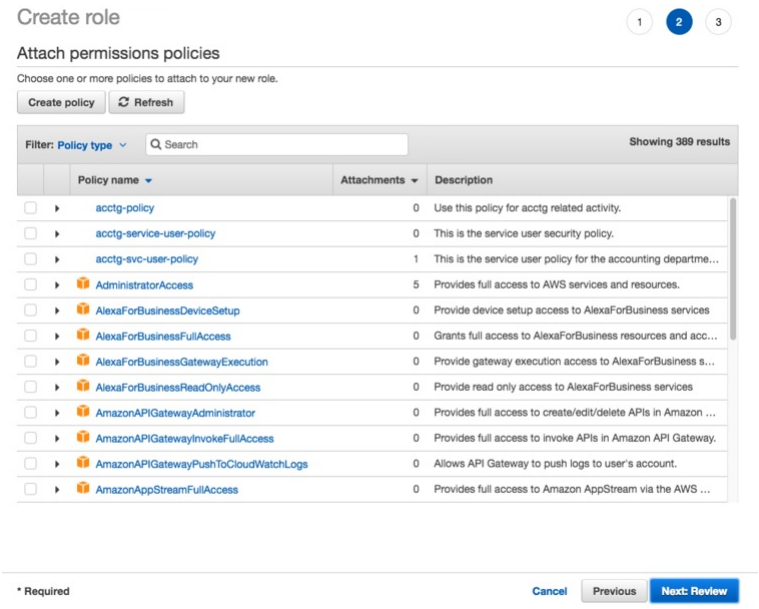


Fig. 6.27: The Attach permissions policies dialog

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

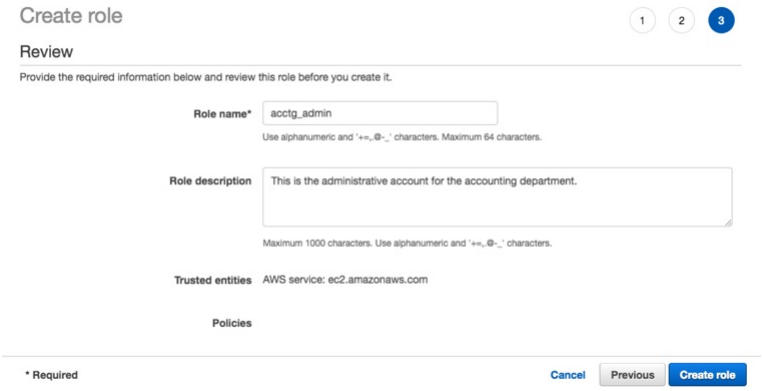


Fig. 6.28: The Review dialog

Use the `Review` dialog 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.

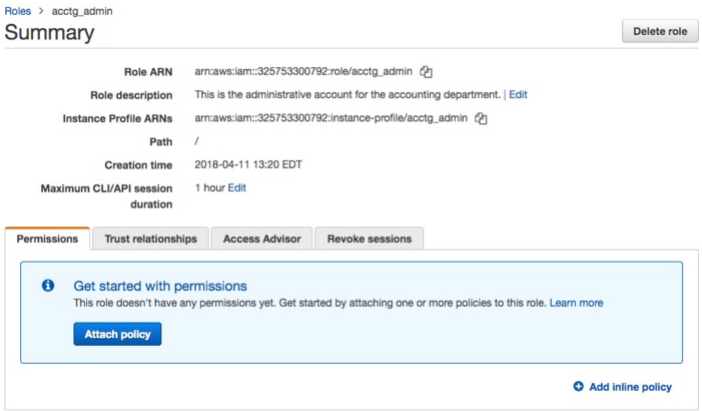


Fig. 6.29: The Summary dialog

The *Summary* dialog will display a Role ARN, but the ARN will not be enabled until the security policy and trust policy are

updated.

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` link to add a security policy. Then, copy the permission policy text into the `JSON` tab.

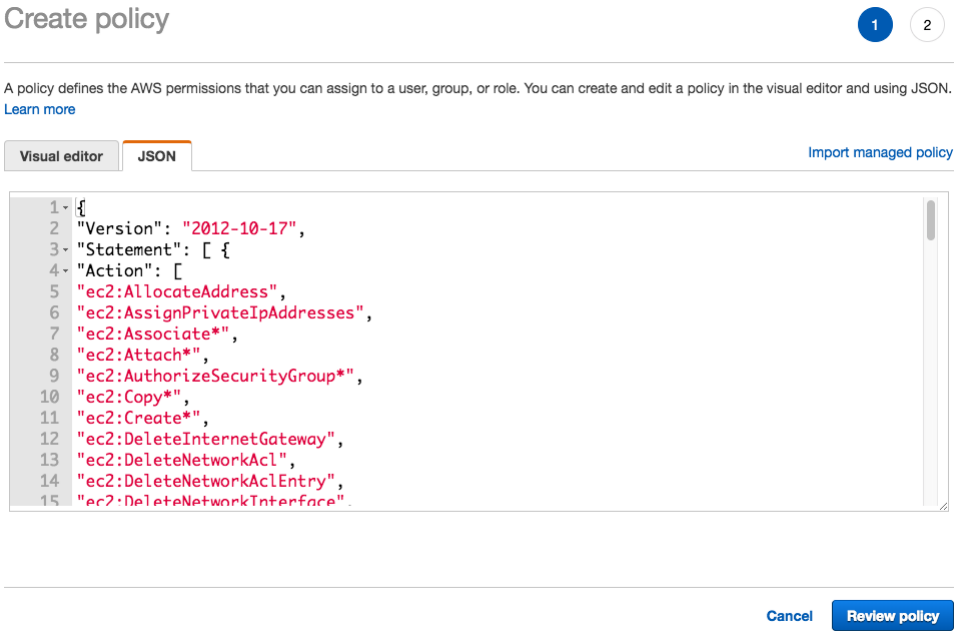


Fig. 6.30: Adding a security policy

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.

Select the `Trust relationships` tab, and click the `Edit trust relationship` button to update the trust relationship assigned to the role.

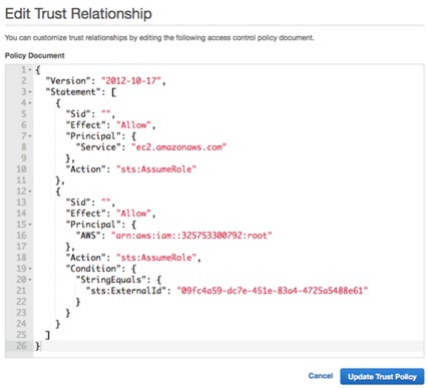


Fig. 6.31: Editing the trust relationship

Replace the displayed content with the provided policy document.

!!! 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.

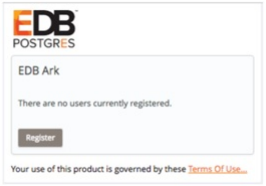


Fig. 6.32: Accessing the New User Registration dialog

Click the `Register` button to open the `New User Registration` dialog.

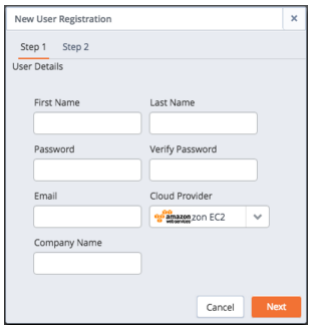


Fig. 6.33: The New User Registration dialog

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.

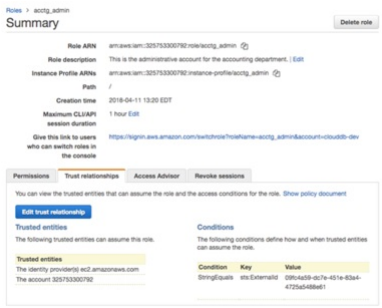


Fig. 6.34: The Summary tab of the Role detail panel

Your Amazon IAM role ARN is displayed on the Amazon role detail panel.

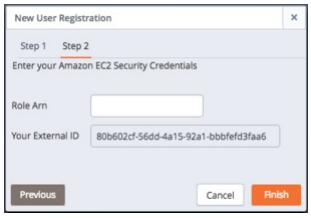


Fig. 6.35: Registering a user on an Amazon EC2 cloud

Enter your Amazon IAM role ARN in the `Role Arn` field on the `Step 2` dialog, and click `Finish` to complete the registration. 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.

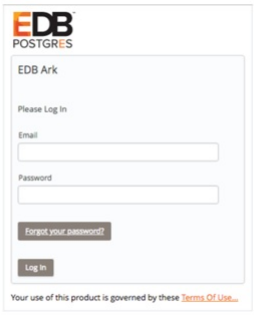


Fig. 6.36: The Login/Register dialog

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.

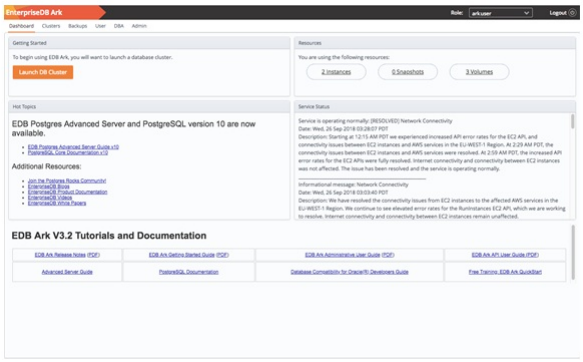


Fig. 6.37: The Ark dashboard tab

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 Creating a Server Image.
2. Use the Ark console to create database engine definitions. For detailed information about defining a database engine, see Creating a Database Engine

1.6.2 Installing the Ark Console on 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.` Create an Azure network security group .

Create an Azure storage account. Launch a VM Image that contains the Ark console.

Configure the Ark console. Register an Ark Administrative user.

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 scale 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 `scale` field for the specified user(s).

For more information about delegating Azure permissions, please see the Azure documentation.

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

Rule Type	Direction	Port	Remote	CIDR Address
All ICMP SSH HTTP HTTPS	Ingress		CIDR CIDR CIDR CIDR	0.0.0.0/0 0.0.0.0/0 0.0.0.0/0 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.

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.

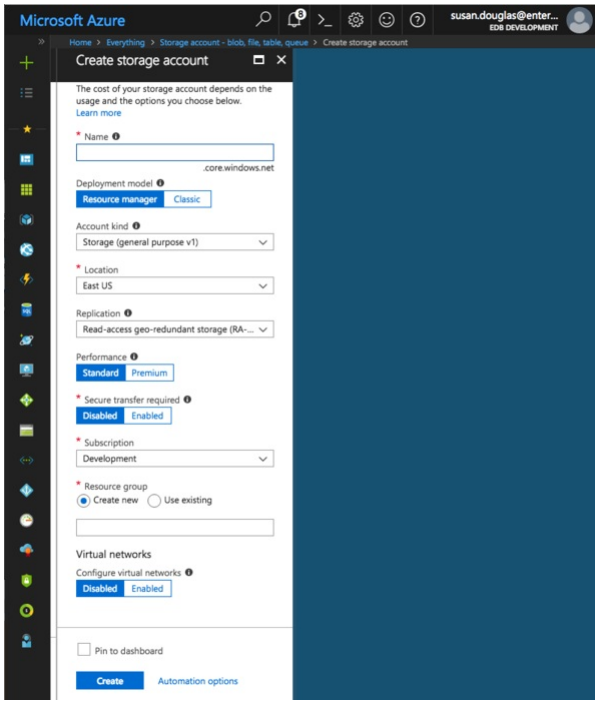


Fig. 6.38: Defining a storage account

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.

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 consult the Azure documentation.

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.

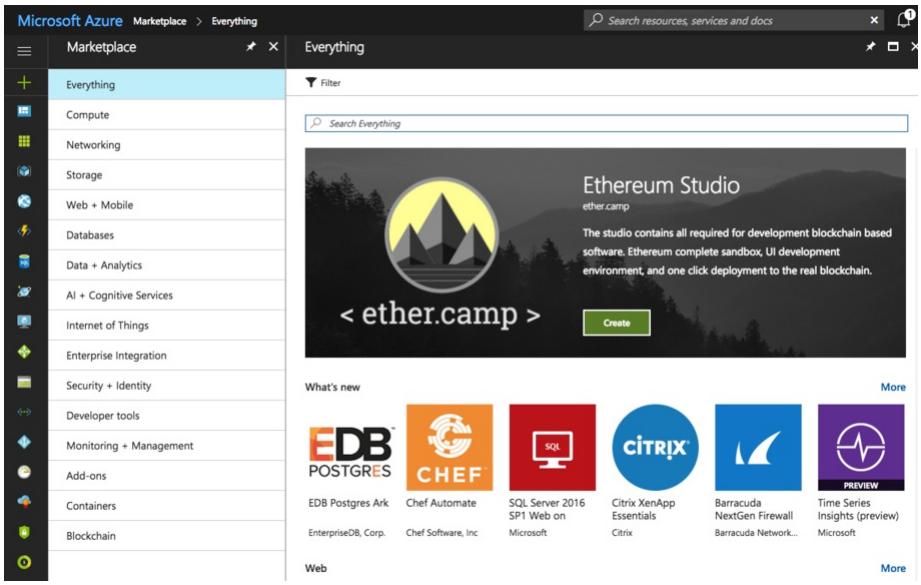


Fig. 6.39: Selecting an image

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.

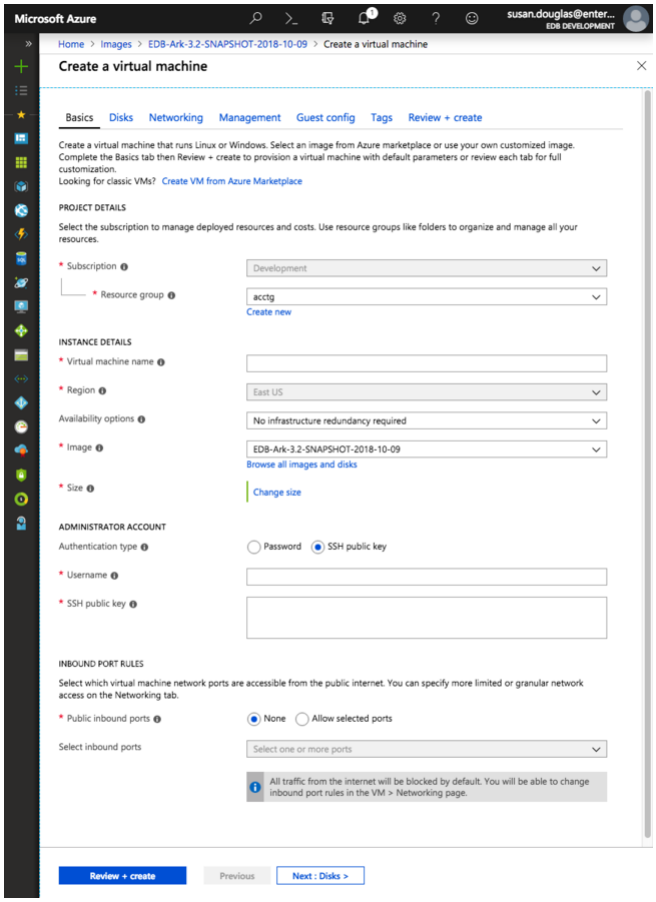


Fig. 6.40: Creating a virtual machine

Use fields on the `Basics` panel 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 and select the machine configuration.

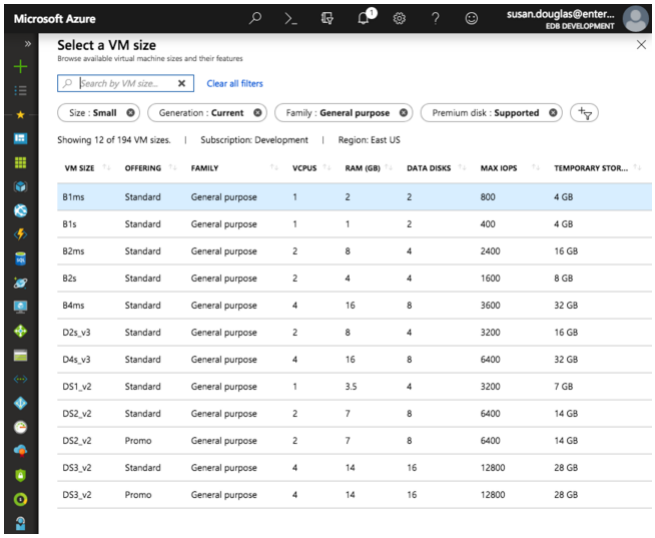


Fig. 6.41: Selecting a machine size

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 extension` panel to identify the script:

- Use the `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`.

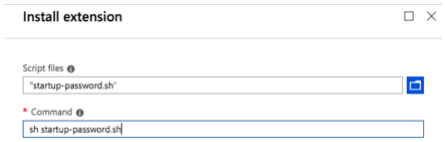


Fig. 6.42: Installing an extension

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.

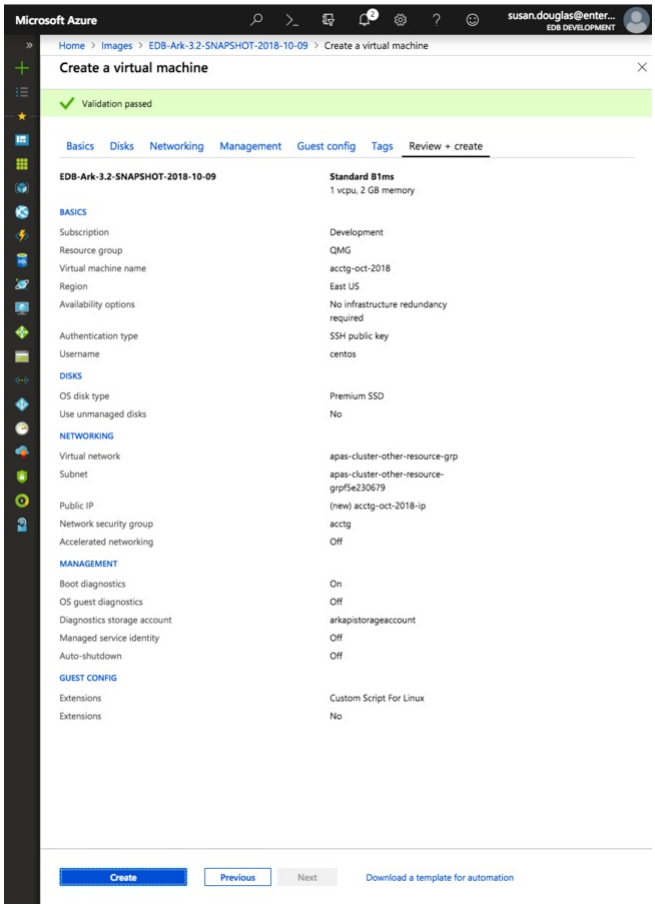


Fig. 6.43: Validating the machine definition

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.

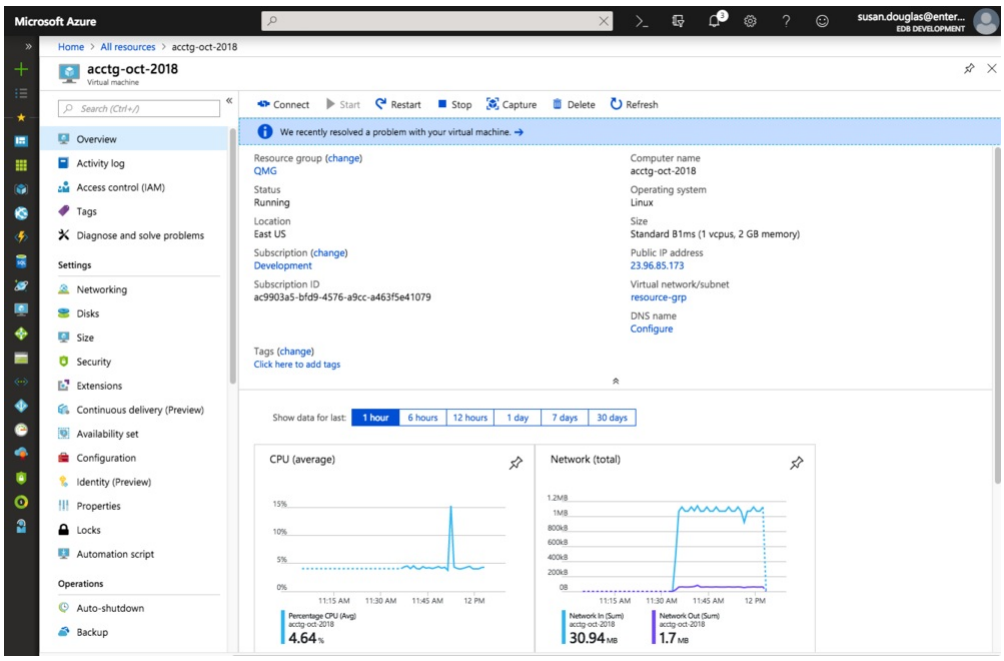


Fig. 6.44: The virtual machine details page

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.

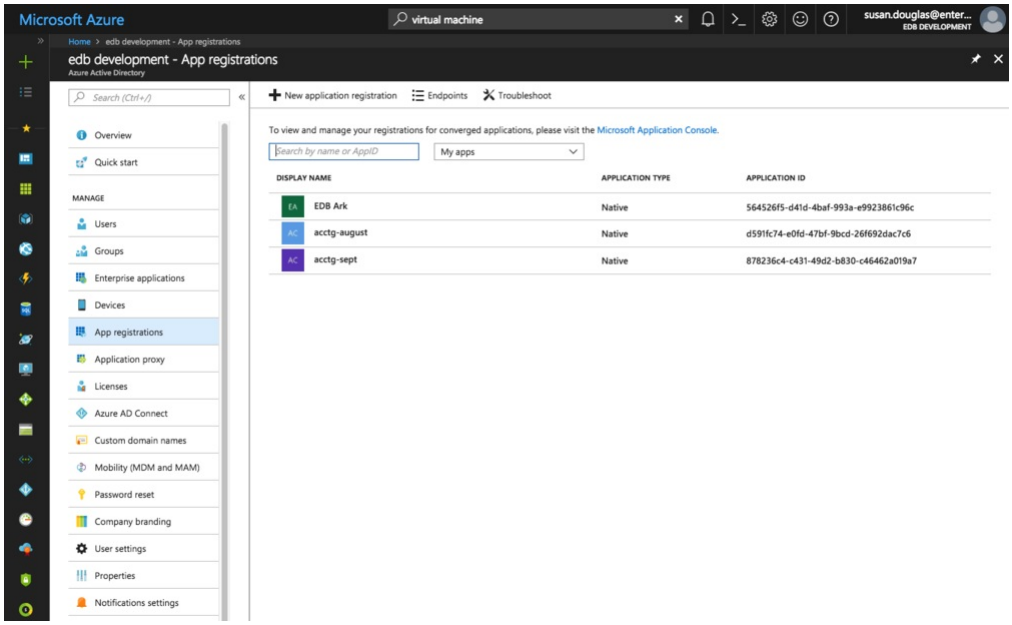


Fig. 6.45: The New application registration page

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.

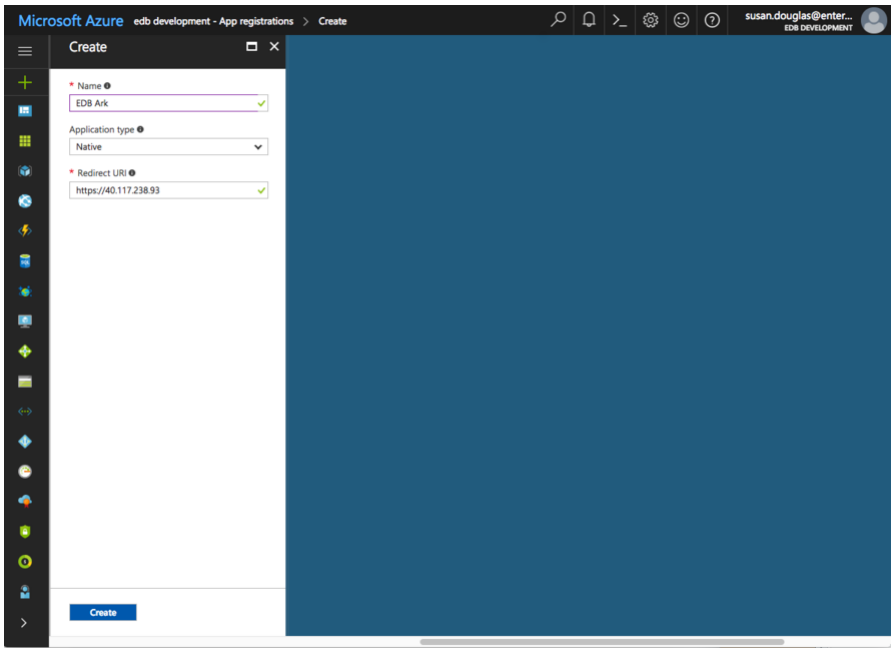


Fig. 6.46: The Create panel

Use fields on the `Create` panel 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.

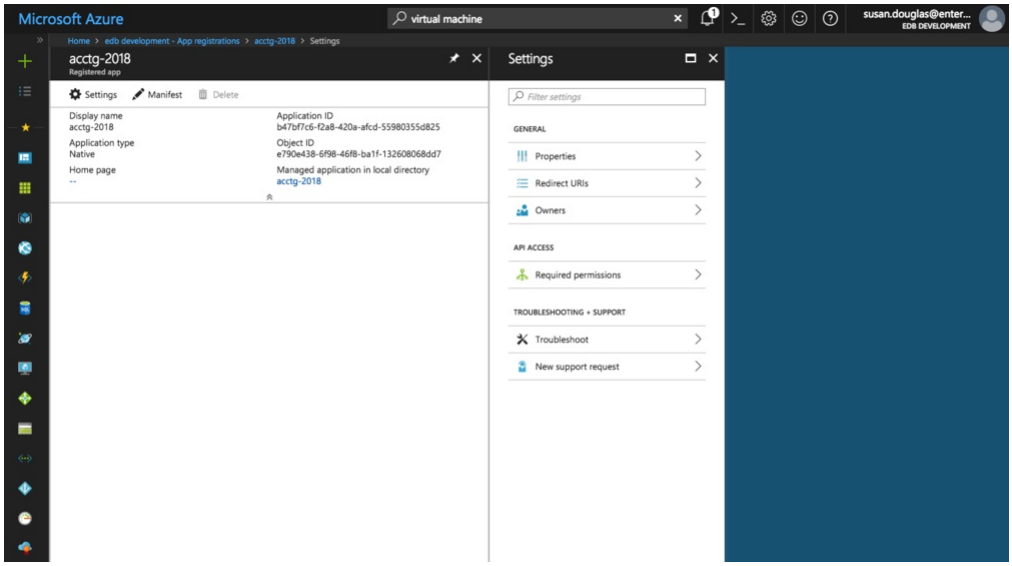


Fig. 6.47: Accessing the Required permissions page

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`.

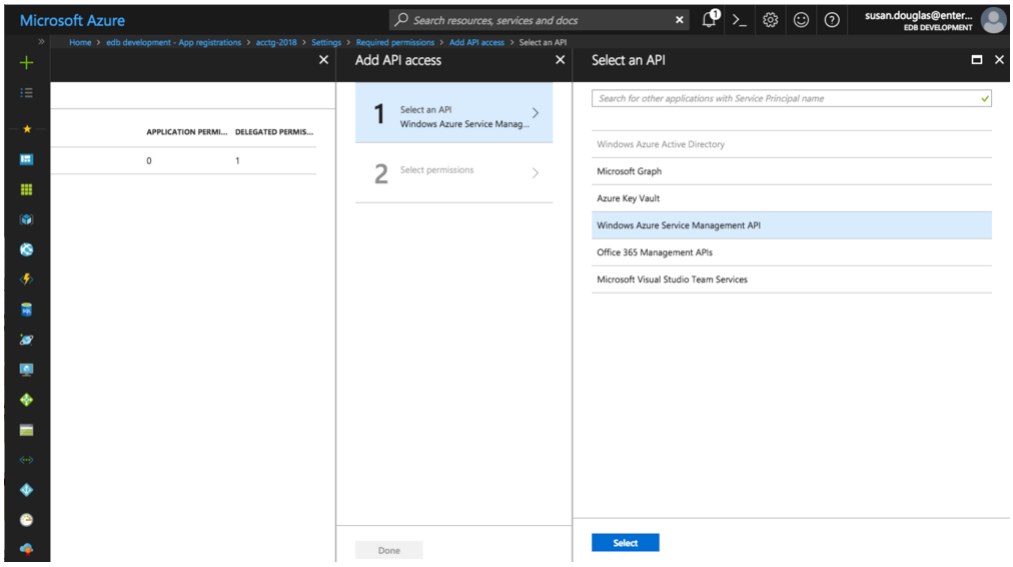


Fig. 6.48: Selecting an API

Click `Select an API`, and then highlight Windows Azure Service Management API.

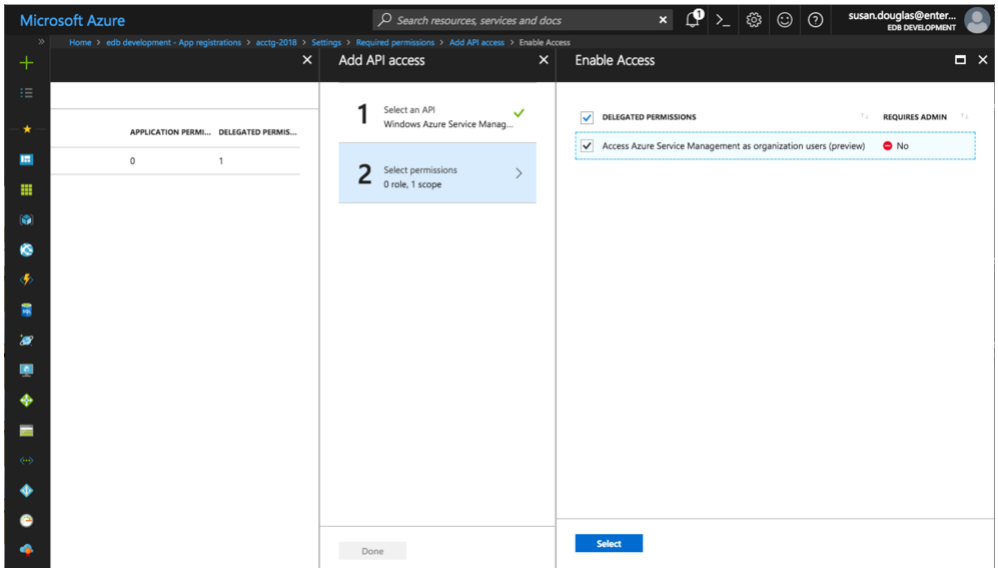


Fig. 6.49: Specifying API permissions

Click `Select permissions`, and then `Access Azure Service Management`; then, click `Select`.

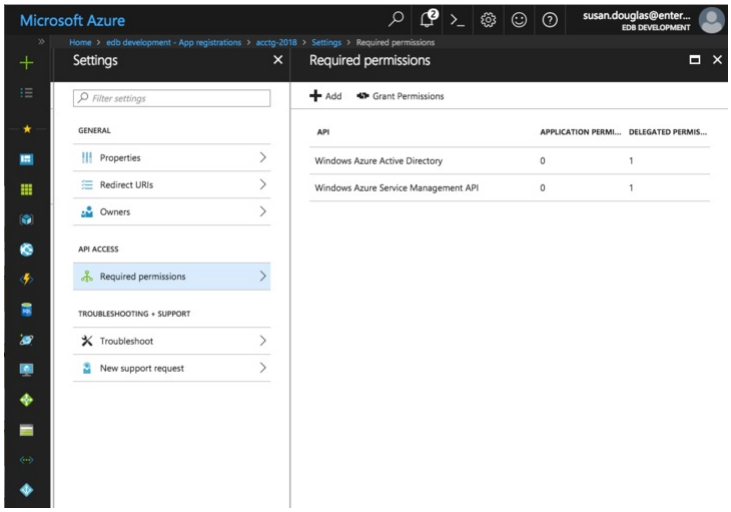


Fig. 6.50: Confirming that the permissions are added

Then, click `Grant Permissions`.

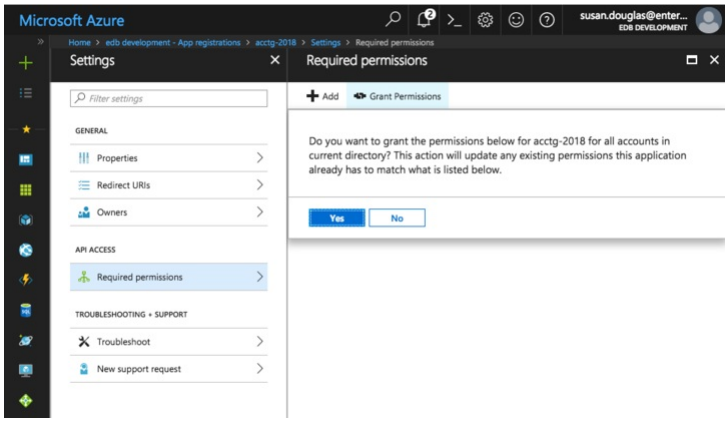


Fig. 6.51: Granting permissions for the applications

When prompted, click `Yes` to confirm that you wish to grant access permissions.

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

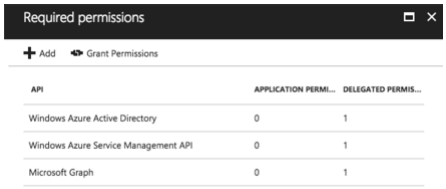


Fig. 6.52: Sufficient resource permissions

When you're finished granting permissions, the `Required permissions` list should include:

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

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.

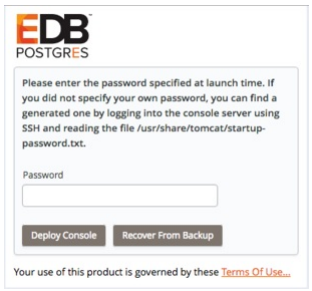


Fig. 6.53: Logging in to the instance

When prompted, provide the console password. 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.

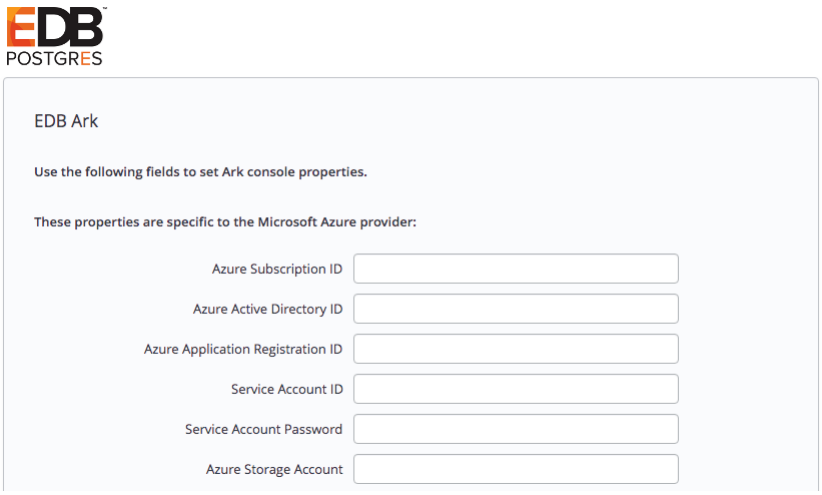


Fig. 6.54: The platform specific property fields

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.

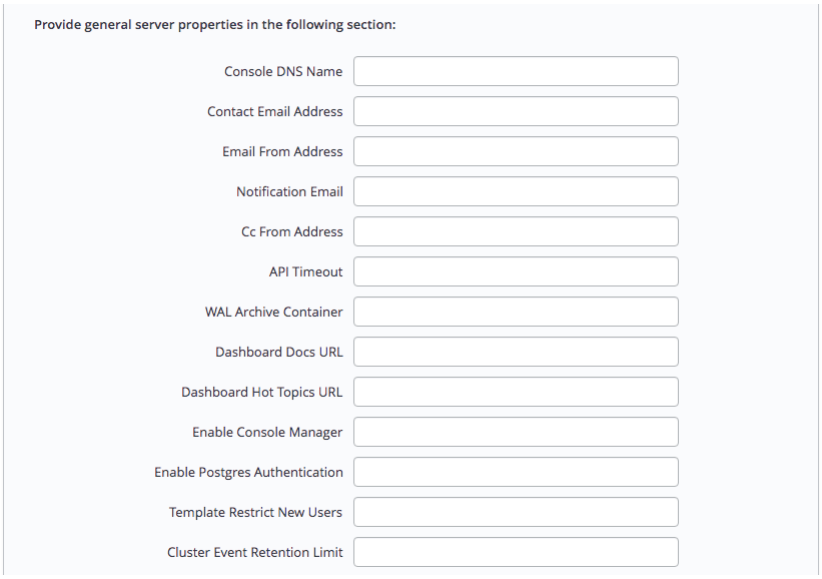


Fig. 6.55: The general server property fields

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 Manager` field to indicate if the console should display console manager functionality. When set to `true`, the console will display the manager; when `false`, the manager will not be displayed.
- 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 set to `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.
- Use the `Cluster Event Retention Limit` field to specify how long the console will keep events for deleted clusters. The default value is `14` days.

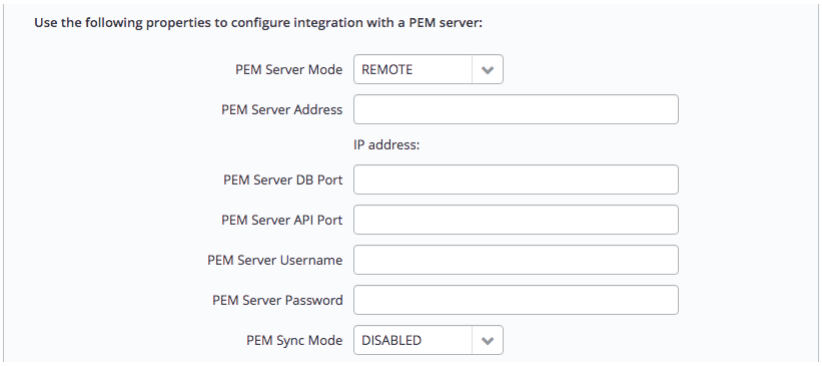


Fig. 6.56: The PEM server property fields

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 IP address of the PEM server host in the `PEM Server IP 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.
- Use the `PEM Synchronization Interval` field to specify the number of minutes between attempts to synchronize the Ark console with the PEM server.

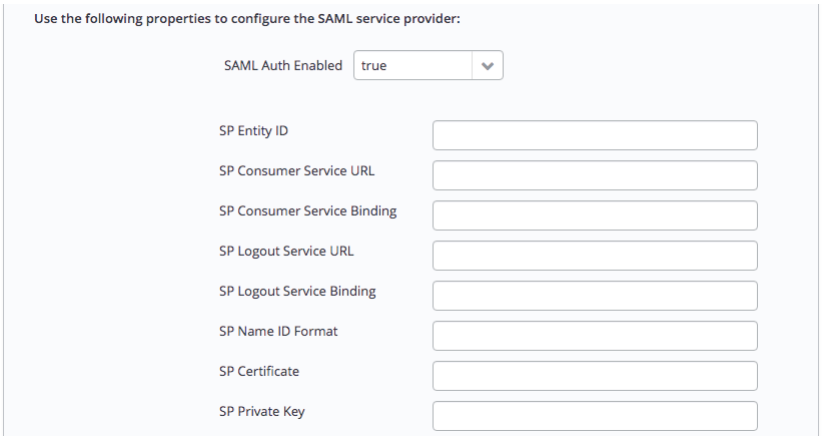


Fig. 6.57: Provide information about the service provider

If you specify `true` in the `SAML auth enabled` field, the Ark console will display the properties required to use SAML authentication when connecting to the Ark console. Use fields on the deployment dialog to specify SAML authentication properties. Use fields in the next section to provide information about the service provider:

- Use the `SP Entity ID` field to provide a URI that specifies the identifier of the service provider.
- Use the `SP Consumer Service URL` field to specify the URL from which the response from the identity provider will be returned.
- Use the `SP Consumer Service Binding` field to specify the SAML protocol binding to be used when returning the response message from the identity provider.

- Use the `SP Logout Service URL` field to specify the URL to which the service provider will specify information about where and how the `Logout Response` message MUST be returned to the requester, in this case our service provider.
- Use the `SP Logout Service Binding` field to specify the SAML protocol binding to be used when returning the `LogoutResponse` or sending the `LogoutRequest` message.
- Use the `SP Name ID Format` field to specify the constraints on the name identifier that will be used to represent the requested subject.
- Use the `SP Certificate` field to specify certificate information; this is usually `x509cert`. The private key of the service files are provided by files placed in the `certs` folder.
- Use the `SP Private Key` field to specify the location of the service providers private key; this must be in `PKCS#8` format.

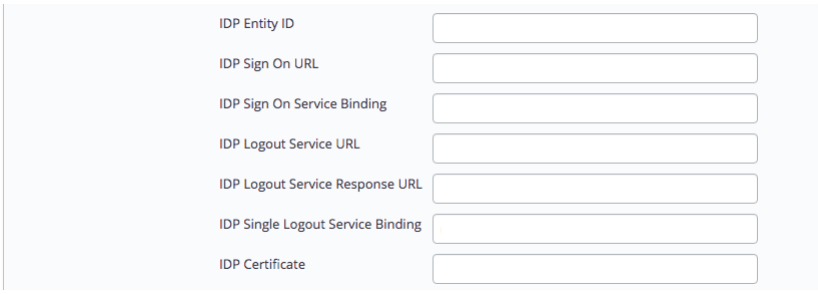


Fig. 6.58: Provide information about the identity provider

Use fields in the next section to provide information about the identity provider:

- Use the `IDP Entity ID` field to specify the identifier of the identity provider (this must be a URI).
- Use the `IDP Sign On URL` field to specify the URL target of the identity provider (where the service provider will send the Authentication Request Message).
- Use the `IDP Sign On Service Binding` field to specify the SAML protocol binding to be used when returning the response message. This version of Ark only supports the HTTP-Redirect binding.
- Use the `IDP Logout Service URL` field to specify the URL Location of the identity provider to which the service provider will send a single logout Request.
- Use the `IDP Logout Service Response URL` field to specify the URL Location of the identity provider to which the service provider will send a single logout response.
- Use the `IDP Single Logout Service Binding` field to specify the SAML protocol binding to be used when returning the response message.
- Use the `IDP Certificate` field to specify the Public x509 certificate of the identity provider.

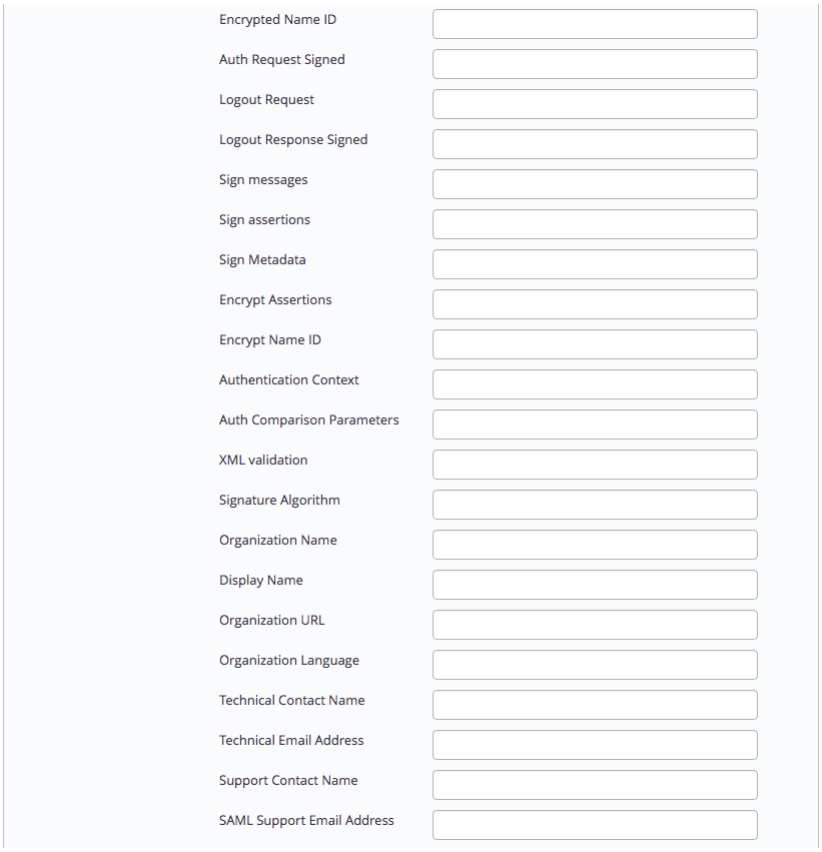


Fig. 6.59: Provide information about your SAML preferences

Use fields in the next section to provide your SAML preferences:

- Use the `Encrypted Name ID` field to indicate that the name identifier of the `samlp:logoutRequest` sent by this service provider will be encrypted; specify `true` or `false`.
- Use the `Auth Request Signed` field to indicate if the `samlp:AuthnRequest` messages sent by this service provider will be signed; specify `true` or `false`.
- Use the `Logout Request` field to indicate if the `samlp:logoutRequest` messages sent by the service provider will be signed; specify `true` or `false`.
- Use the `Logout Response Signed` field to indicate if the `samlp:logoutResponse` messages sent by the service provider will be signed; specify `true` or `false`.
- Use the `Sign messages` field to sign the metadata. If you leave the field empty, the metadata will not be signed. If you wish to provide a signature, provide a comma separated `keyFileName`, `certFileName` pair.
- Use the `Sign assertions` field to indicate a requirement for the `samlp:Response`, `samlp:LogoutRequest`, and `samlp:LogoutResponse` elements received by the service provider to be signed; specify `true` or `false`.
- Use the `Sign Metadata` field to indicate that the metadata of this service provider must be signed; specify `true` or `false`.
- Use the `Encrypt Assertions` field to indicate that the assertions received by this service provider must be encrypted; specify `true` or `false`.
- Use the `Encrypt Name ID` field to indicate that the name identifier received by this service provider must be encrypted; specify `true` or `false`.
- Use the `Authentication Context` field to specify that `Set Empty` and `no AuthContext` will be sent in the `AuthNRequest`. You can set multiple values in a comma-delimited list.
- Use the `Auth Comparison Parameters` field to specify that the `authn` comparison parameter to be set; this field defaults to `exact`.
- Use the `XML validation` field to indicate if the service provider will validate all received xmls; specify `true` or `false`.
- Use the `Signature Algorithm` field to specify the algorithm that the toolkit will use for the signing process. Specify one of the following:

- `http://www.w3.org/2000/09/xmldsig#rsa-sha1`
- `http://www.w3.org/2000/09/xmldsig#dsa-sha1`
- `http://www.w3.org/2001/04/xmldsig-more#rsa-sha256`
- `http://www.w3.org/2001/04/xmldsig-more#rsa-sha384`
- `http://www.w3.org/2001/04/xmldsig-more#rsa-sha512`

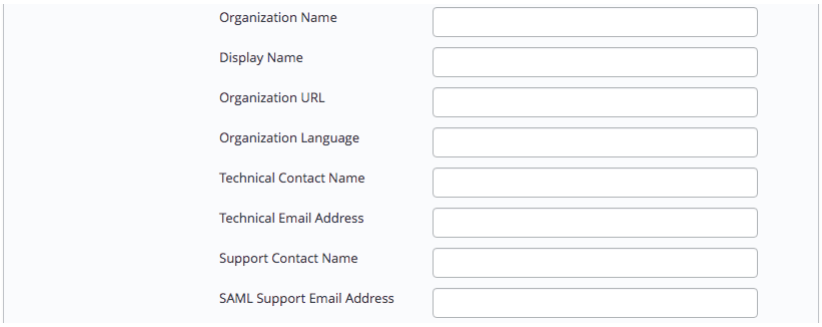


Fig. 6.60: Provide information about your organization

Use fields in the next section to provide information about your organization:

- Use the `Organization Name` field to specify the name of the organization for which authentication is being provided.
- Use the `Display Name` field to specify the display name of the organization.
- Use the `Organization URL` field to specify the URL of the organization.
- Use the `Organization Language` field to specify the primary language used by the organization.
- Use the `Technical Contact Name` field to specify the name of a technical contact.
- Use the `Technical Email Address` field to specify a contact email address for the technical contact.
- Use the `Support Contact Name` field to specify the name of a support contact.
- Use the `SAML Support Email Address` field to specify the email address of the SAML support contact.

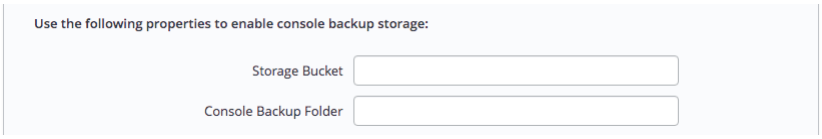


Fig. 6.61: The console backup storage fields

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 consult the Azure documentation.

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

Fig. 6.62: 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.

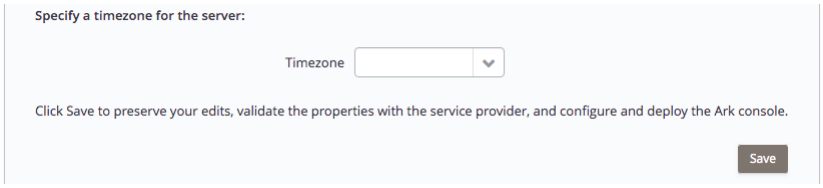


Fig. 6.63: The timezone field

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.

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.

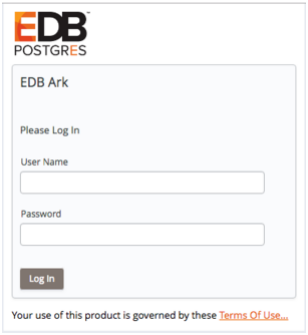


Fig. 6.64: The Login dialog

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.

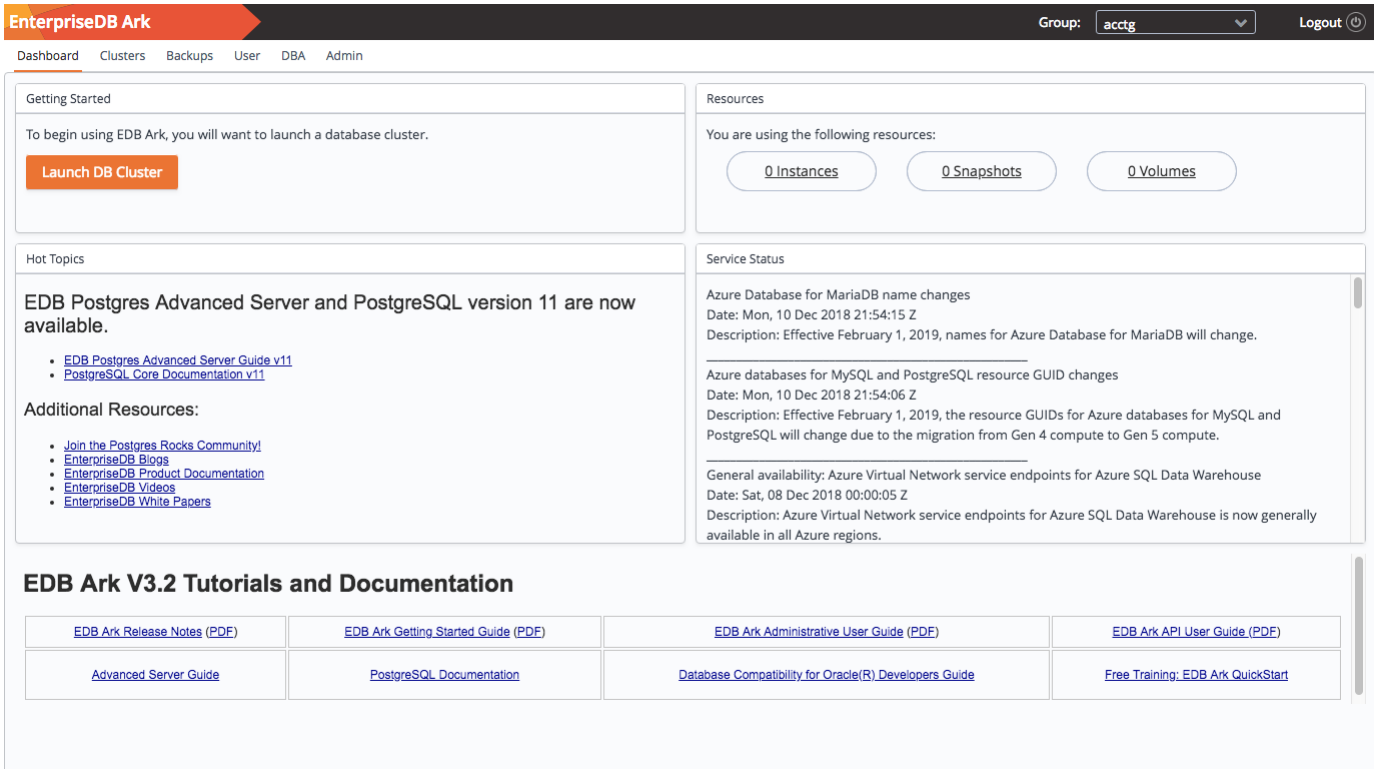


Fig. 6.65: The Dashboard of the EDB Ark Administrator's console

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`, available on the Ark `Dashboard`.
- Use the Admin tab to create the server images and database engines that will be used by non-administrative users.

1.7 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.

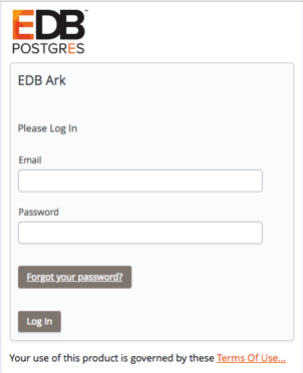


Fig. 7.1: The Login dialog

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 below.

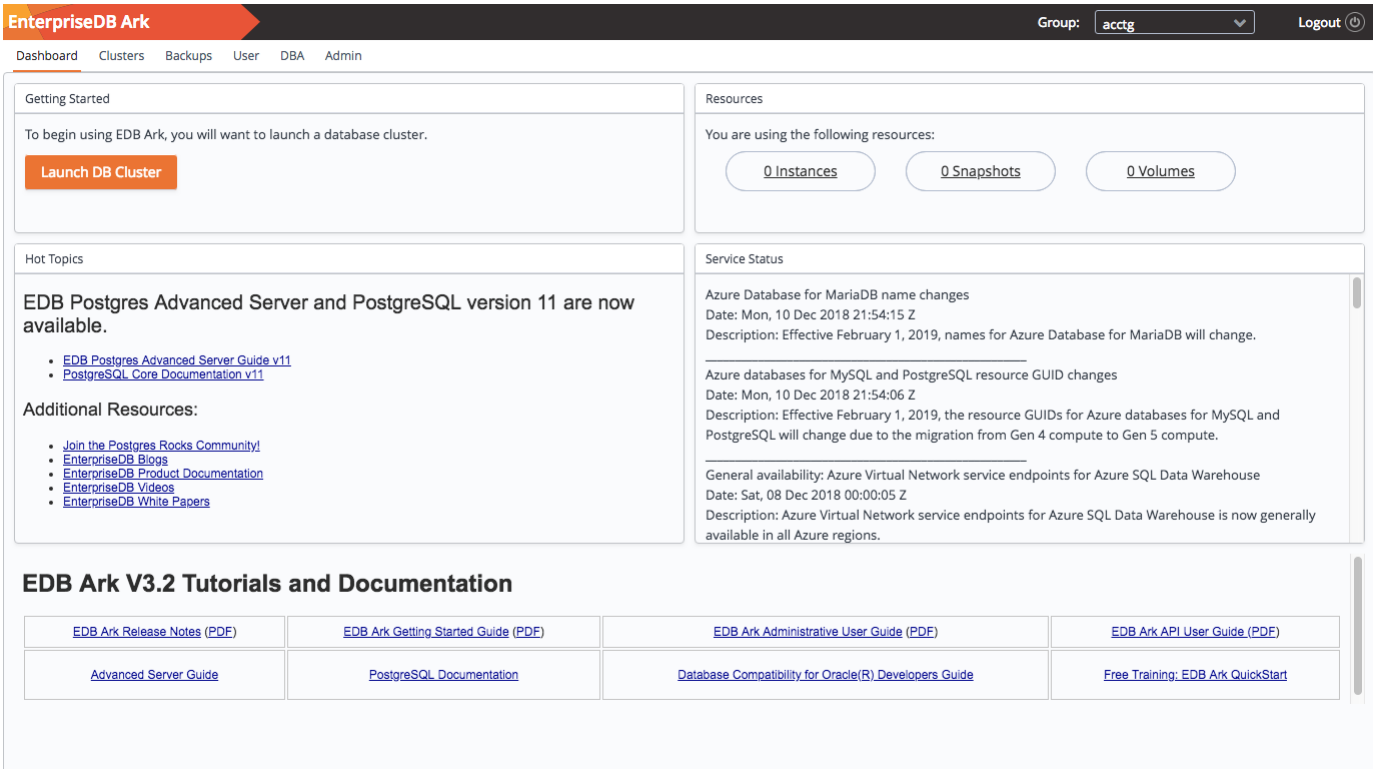


Fig. 7.2: The EDB Ark Administrator's console

1.8 Using the Admin Tab

Use options on the `Admin` tab to perform administrative tasks such as creating federations of consoles, managing servers and database engines, and defining templates. You can also use the `Admin` tab to manage users and deployment properties.

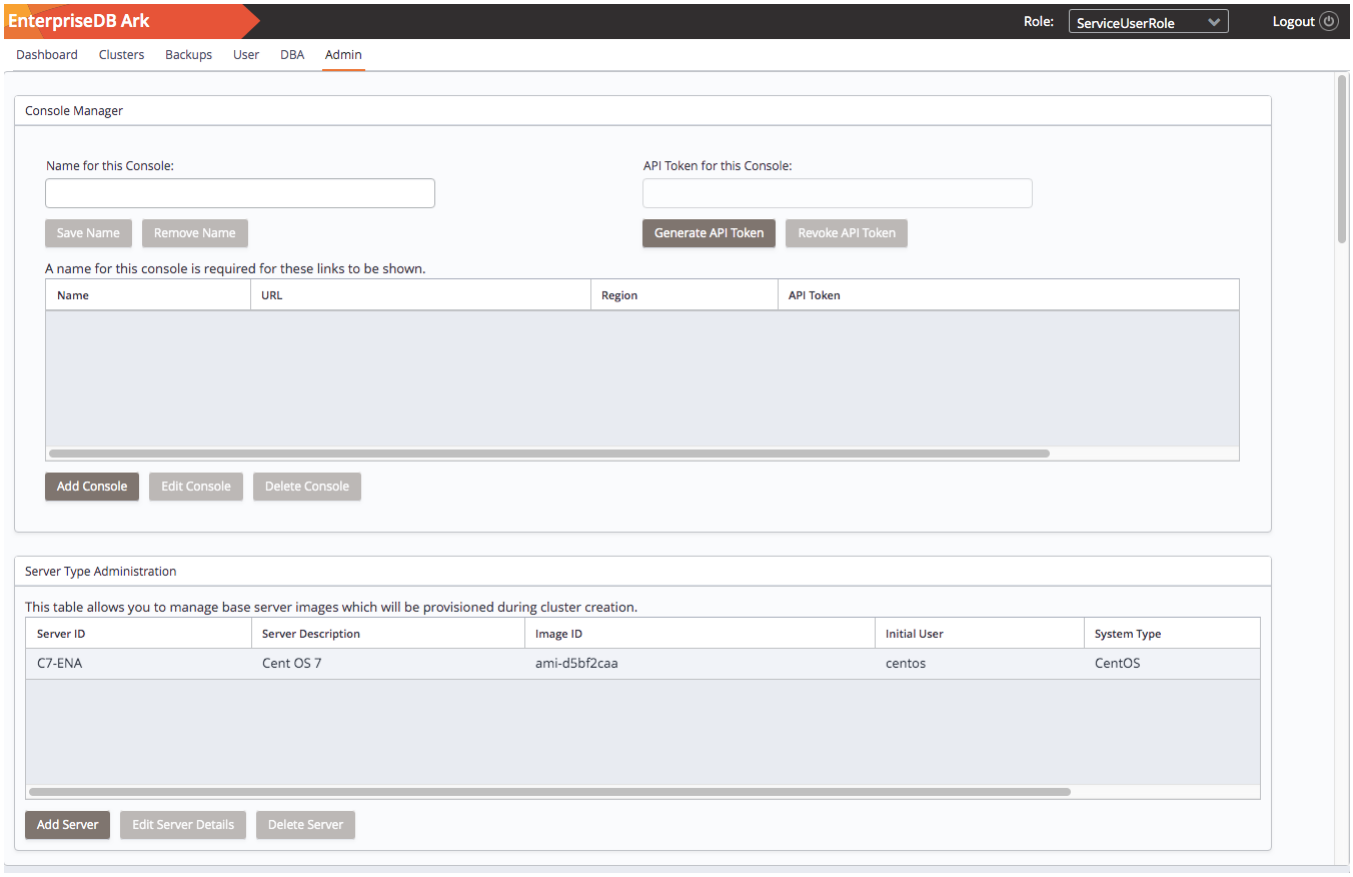


Fig. 8.1: The Admin tab.

Console Manager

Use the fields in the `Console Manager` panel to:

- Make a console available through the `Consoles` drop-down navigation listbox on the Ark console.
- Generate a token that can be used to configure the current console as part of a *federation* of Ark consoles. A federated console can deploy resources (such as standby servers) on another console that resides in another region.
- Register other consoles with the current console as part of a *federation* of consoles.

For more information, see Using the Console Manager.

Server Type Administration

A fresh installation of EDB Ark will include default database engine configurations for:

- 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 Managing Server Images.

DB Engine Administration

The databases made available through the `DB Engine Administration` panel 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 Managing Server Images.
- 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 Managing Database Engines.

Template Administration

Use the `Template Administration` panel 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 Template Administration.

RHEL Subscription Management

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

- Add, modify, or delete RHEL subscription information.

For more information, see Red Hat Subscription Management.

IAM Roles Administration (AWS only)

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

- Make Amazon ARNs available for use in Ark user accounts (AWS).

For information about user administration options, see Managing Amazon Roles.

User Administration

Options in the `User Administration` panel 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 User Administration.

Download Console Logs

Click the `Download` button in the `Download Console Logs` panel 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 Accessing the Console Logs.

Download Core Usage Logs

Click the `Download` button in the `Download Core Usage Logs` panel to download a zip file that contains the usage logs for the underlying server.

For more information, see Accessing the Core Usage Logs.

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.

For more information, see Taking 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 Editing Installation Properties.

Using the Console Manager

Use fields in the `Console Manager` panel to manage the names and links displayed by the `Consoles` drop-down listbox on the Ark console. You can also use the Console Manager to manage federated consoles and provide access to cross-region resources. The console manager provides access to optional functionality; if you are not federating consoles or do not wish to use the `Consoles` drop-down, you are not required to complete the fields in the panel.

For more information about creating and using federated consoles, see Using the Console Manager to Create a Federated Console.

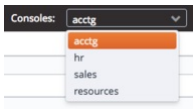


Fig. 8.2: The Consoles drop-down.

The `Consoles` drop-down listbox (in the upper-right corner of the Ark console) provides quick access to other consoles. When you select a name from the `Consoles` drop-down listbox, the Ark console opens a browser tab and navigates to the address associated with the name. Use the `Console Manager` section of the `Admin` tab to manage the console names and addresses that are displayed in the `Consoles` drop-down.

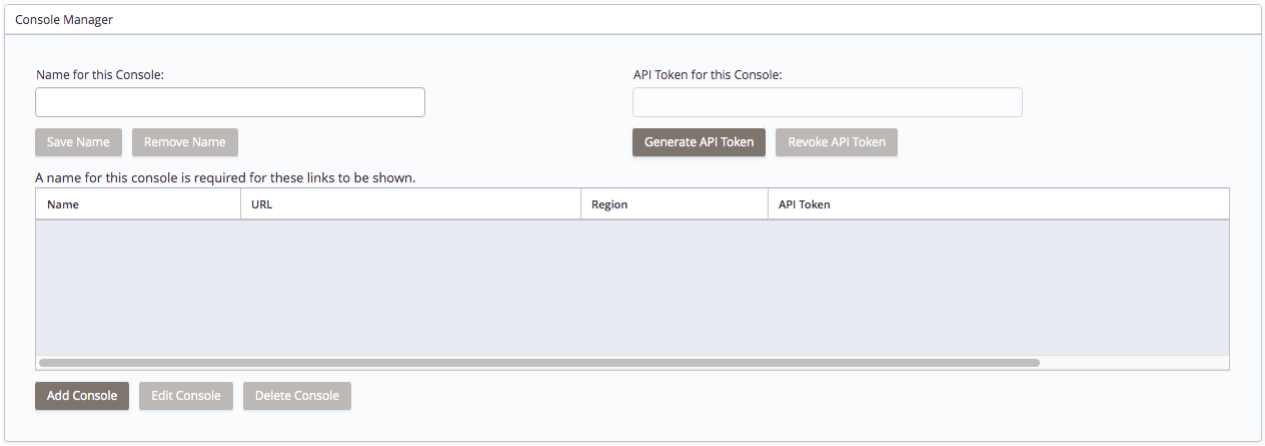


Fig. 8.3: The Console Manager panel

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.

Fig. 8.4: Provide a name for the console

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.

Fig. 8.5: Saving the console name.

Then, use the `Add Console` 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.
- Optionally, provide the API token associated with the console in the `API Token` field. This field is only required when you are federating consoles.

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.

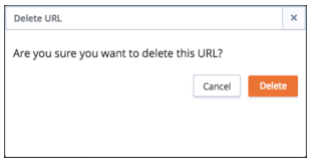


Fig. 8.6: Deleting a console.

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 Manager` table, or click `Cancel` to exit the dialog without deleting the entry.

Using the Console Manager to Create a Federated Console

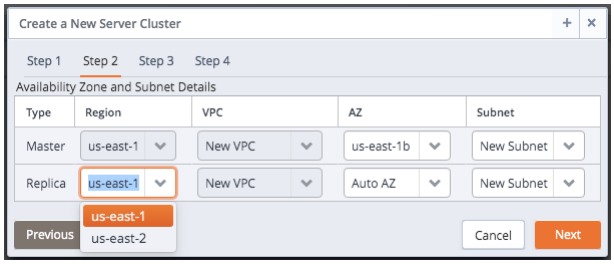
You can use fields in the `Console Manager` panel to generate and share access tokens between consoles to create a *federation* of consoles. After creating a federation, you can create or clone cluster members in any region in which one of the federated consoles resides. A federated console may have resources that reside in more than one region; standby nodes that reside in other regions will reflect the state of the master node.

!!! Note Each federation may only contain one console from a specific region; if you try to add more than one console from a given region to a federation, Ark will return an error.

When creating a federation, please note:

- All of the federated consoles must be configured with the same set of database engines and server images:
    - Each database engine must have the same configuration across all federated consoles (including the engine id and name).
    - Each server image must have the same configuration across all federated consoles; the image id will vary by region.

For example, you might have two consoles; one that resides in Region `us-east-1` and one that resides in Region `us-east-2`. You can create a federation of those consoles by *registering* the consoles with each other. After registering the consoles (sharing the URL and token of each console within the federation with the other console), you will be allowed to select the `Region` in which replica nodes are created or cloned.



!!! Note Standby nodes that reside in a different region than the master console will be used for load balancing. Failover to nodes that reside on a different region than the master node is not supported. If a master node fails and you do not have a standby node in the same region as the master node, Ark will create a replacement node in the same region as the original master.

All of the resources that reside on federated consoles will be visible on the `Clusters` tab of all of the consoles within the federation. To federate consoles you must:

1. Use the `Name for this Console` field to create a name for each console that will be federated. Click within the field and provide a console name; when finished, click the `Save Name` button. The console name will be displayed on the title bar of the Ark console.

Fig. 8.8: Providing a console name.

2. Click the `Generate API Token` button to create a token for the console. The API token is displayed in the `API Token for this Console` field.

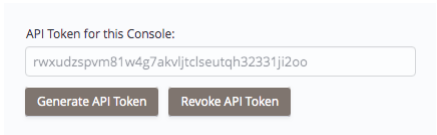


Fig. 8.9: Generating an API token.

3. Use the `Add Console` dialog on each console within the federation to register the other consoles with which they will be federated.

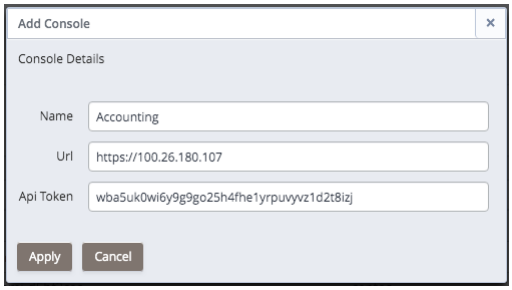


Fig. 8.10: Registering a console.

After registering each with all of the other consoles within the federation, you will be allowed to create resources in any region in which the federated consoles reside.

Revoking a Token

Select the `Revoke API Token` button to revoke the token assigned to the console.

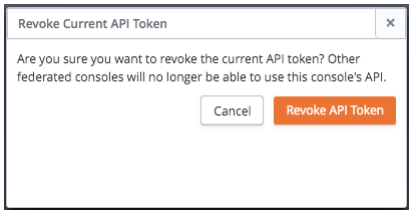


Fig. 8.11: Confirm you wish to revoke the token.

If the console is a member of a federation, the other members of the federation will no longer be able to create resources in the area in which the console resides. To restore membership in a federation, you must generate a new token for the console and edit the console definitions of the other federation members, updating the token associated with the console.

After creating a federation, you can:

- Log in to any federated console and initiate a cluster backup.
- Log in to any federated console and delete a cluster.
- Initiate a clone operation regardless of the region in which the node resides.
- Log in to any federated console and initiate a yum update on a cluster.
- Scale a federated cluster in multiple regions.
- Perform machine scaling on a cluster that resides on a federated console.

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` panel to manage server images.

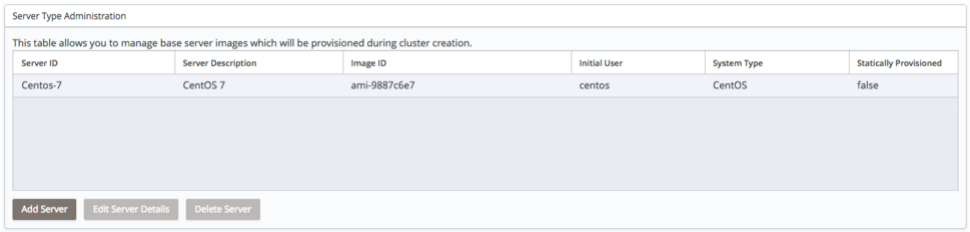


Fig. 8.12: The Server Type Administration section of the Admin tab

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 opens.

Fig. 8.13: 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. Review the Azure documentation for information about downloading and installing the 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

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

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, review the AWS documentation.

- 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

Click the `Edit Server Details` button to open the `Edit Server Details` dialog and modify the properties of a server.

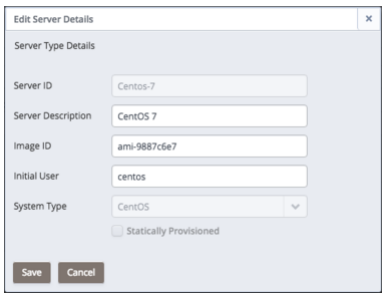


Fig. 8.14: The Edit Server dialog

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.

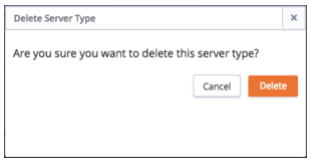


Fig. 8.15: The Delete Server Type dialog

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

Fig. 8.16: 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; before removing the server, you must delete any dependent engines.

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 role or group will be made available to all users with access to that group. You can use the `DB Engine Administration` section of

the Admin tab to create and manage database engines.

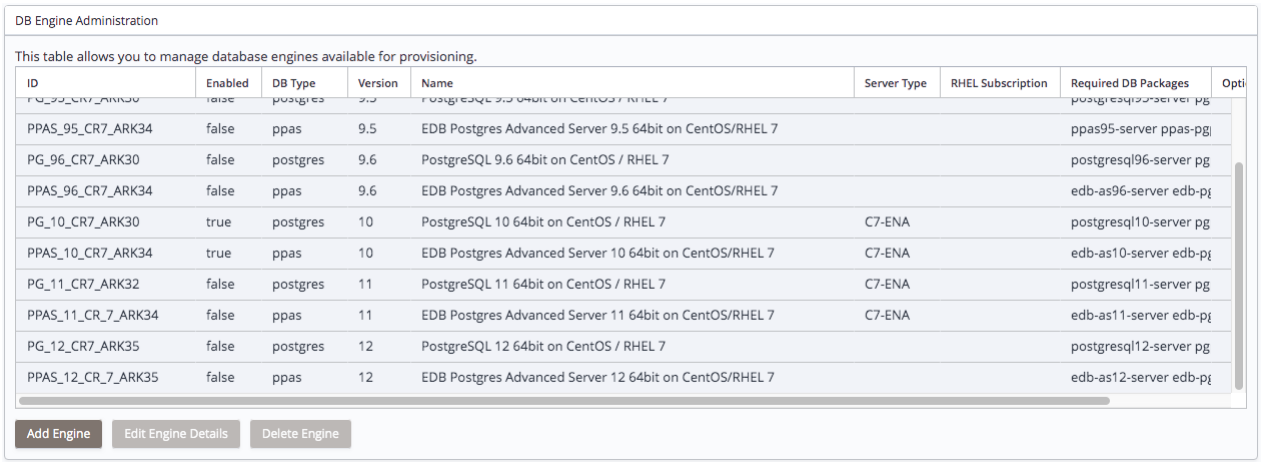


Fig. 8.17: The DB Engine Administration section of the Admin tab

Please note: The EnterpriseDB repository structure has changed; after upgrading your Ark console to v3.5, you will have a combination of old and new DB engine configurations in the Ark Admin panel. You should migrate to the new DB engines for all new clusters since these contain the proper EDB YUM repository configuration.

The Ark console ships with a number of default engine definitions. Before using an engine, you must create a server (see Managing Server Images) 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. Engine definitions may 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; for credentials, visit the EnterpriseDB website.

PostgreSQL 9.4 64bit on CentOS / RHEL 7

```
https://yum.postgresql.org/9.4/redhat/rhel-7-x86_64/pgdg-redhat-repo-
latest.noarch.rpm
https://USERNAME:PASSWORD@yum.enterprisedb.com/edb-repo-rpms/edb-repo-
latest.noarch.rpm

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

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

```
https://USERNAME:PASSWORD@yum.enterprisedb.com/edb-repo-rpms/edb-repo-
latest.noarch.rpm

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

PostgreSQL 9.5 64bit on CentOS / RHEL 7

```
https://yum.postgresql.org/9.5/redhat/rhel-7-x86_64/pgdg-redhat-repo-
latest.noarch.rpm
https://USERNAME:PASSWORD@yum.enterprisedb.com/edb-repo-rpms/edb-repo-
latest.noarch.rpm

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

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

```
https://USERNAME:PASSWORD@yum.enterprisedb.com/edb-repo-rpms/edb-repo-
latest.noarch.rpm

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

PostgreSQL 9.6 64bit on CentOS / RHEL 7

```
https://yum.postgresql.org/9.6/redhat/rhel-7-x86_64/pgdg-redhat-repo-
latest.noarch.rpm
https://USERNAME:PASSWORD@yum.enterprisedb.com/edb-repo-rpms/edb-repo-
latest.noarch.rpm

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

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

```
https://USERNAME:PASSWORD@yum.enterprisedb.com/edb-repo-rpms/edb-repo-
latest.noarch.rpm

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

PostgreSQL 10 64bit on CentOS / RHEL 7

```
https://yum.postgresql.org/10/redhat/rhel-7-x86_64/pgdg-redhat-repo-
latest.noarch.rpm
https://USERNAME:PASSWORD@yum.enterprisedb.com/edb-repo-rpms/edb-repo-
latest.noarch.rpm

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

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

```
https://USERNAME:PASSWORD@yum.enterprisedb.com/edb-repo-rpms/edb-repo-
latest.noarch.rpm

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

PostgreSQL 11 64bit on CentOS / RHEL 7

```
https://yum.postgresql.org/11/redhat/rhel-7-x86_64/pgdg-redhat-repo-
latest.noarch.rpm
https://USERNAME:PASSWORD@yum.enterprisedb.com/edb-repo-rpms/edb-repo-
latest.noarch.rpm

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

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

```
https://USERNAME:PASSWORD@yum.enterprisedb.com/edb-repo-rpms/edb-repo-
latest.noarch.rpm

Required Packages: pkgs: edb-as11-server edb-pgpool37 edb-as11-pgpool37-extensions
edb-pem-agent
```

PostgreSQL 12 64bit on CentOS / RHEL 7

```
https://yum.postgresql.org/reporpms/EL-7-x86_64/pgdg-redhat-repo-latest.noarch.rpm
https://USERNAME:PASSWORD@yum.enterprisedb.com/edb-repo-rpms/edb-repo-
latest.noarch.rpm

Required Packages: postgresql12-server pgpool-II-12 pgpool-II-12-extensions edb-pem-
agent
```

EDB Postgres Advanced Server 12 64bit on CentOS / RHEL 7

```
https://USERNAME:PASSWORD@yum.enterprisedb.com/edb-repo-rpms/edb-repo-
latest.noarch.rpm

Required Packages: edb-as12-server edb-pgpool40 edb-as12-pgpool40-extensions edb-
pem-agent
```

Adding, Modifying, or Deleting Engine Definitions

Use the `Add Engine` dialog 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`.

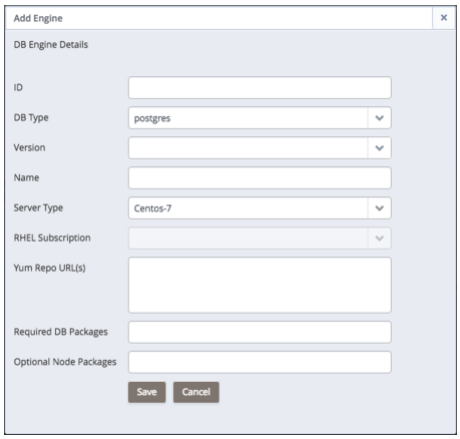


Fig. 8.18: The Add Engine dialog

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. 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. The repository URL should take the form:

  `http://<user_name>:<password>@<repository_url>`

  Where:

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

  `password` specifies the password associated with the repository user. 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 if you need connection credentials for the EnterpriseDB repositories, or visit the EnterpriseDB website.

- 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 an Advanced Server 12 database, the package list must include:

```
edb-as12-server
edb-pgpool40
edb-as12-pgpool40-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
```

For a PostgreSQL 12 database, the package list must include:

```
postgresql12-server
pgpool-II-12
pgpool-II-12-extensions
edb-pem-agent
```

- 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.

Modifying an Engine

To modify an engine, use the `Edit Engine Details` button to open the `Edit Engine Details` dialog.

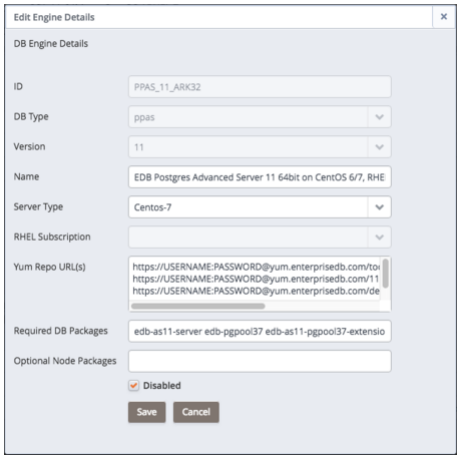


Fig. 8.19: The Edit Engine Details dialog

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.

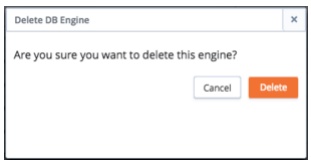


Fig. 8.20: The Delete DB Engine dialog

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.

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`.

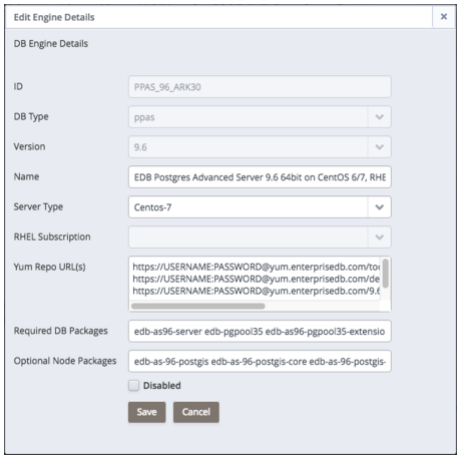


Fig. 8.21: Modifying the Engine Details dialog

When the `Edit Engine Details` dialog opens, 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.

Visit the PostGIS project site for detailed information about using PostGIS.

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` panel to create and manage templates.

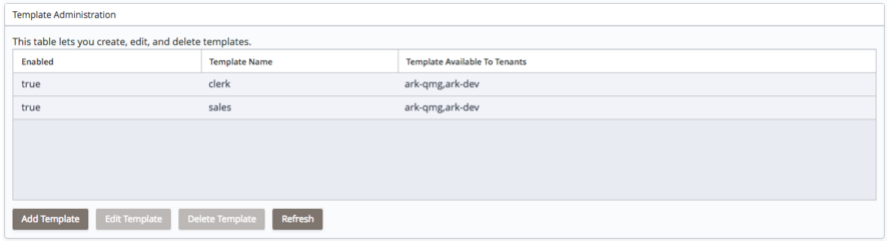


Fig. 8.22: The Template Administration section of the Admin dashboard

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.

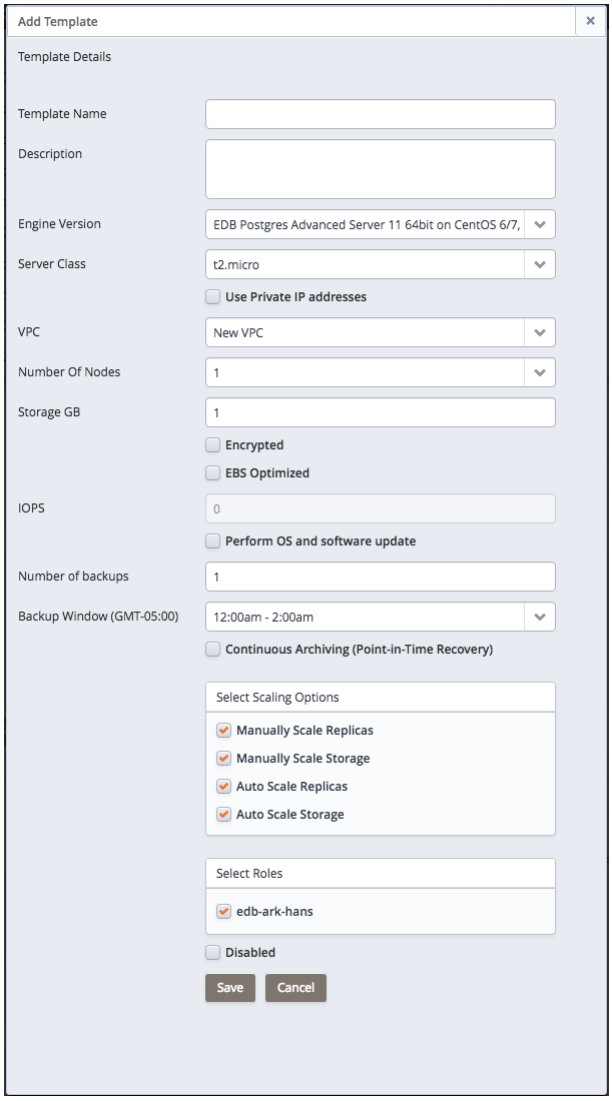


Fig. 8.23: The Template Administration section of the Admin dashboard

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 name in the `Select Roles` box to indicate that members of the selected role 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.

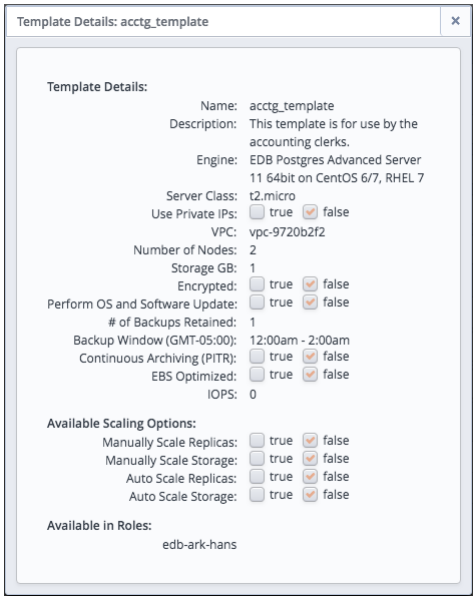


Fig. 8.24: The template details popup

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. Visit the Red Hat website for information about Red Hat Subscription Manager.

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.

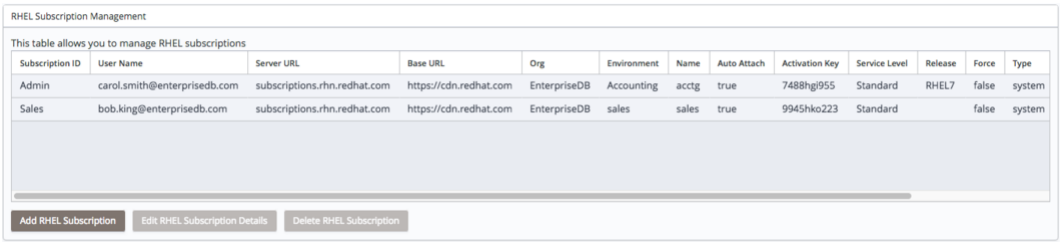


Fig. 8.25: The RHEL Subscription Management section.

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

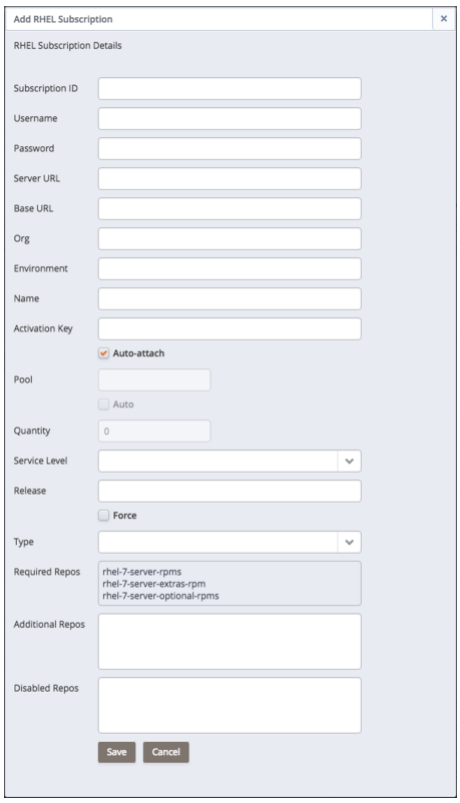


Fig. 8.26: The Add RHEL Subscription dialog

Use fields on the `Add RHEL Subscription` dialog 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.

Fig. 8.27: RHEL Subscription details

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 Managing Database Engines 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.

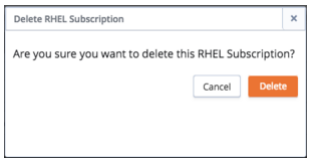


Fig. 8.28: Confirming that you wish to delete a subscription description

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

Managing Amazon Roles

Amazon Role ARNs that are listed in the `IAM Roles Administration` table 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.

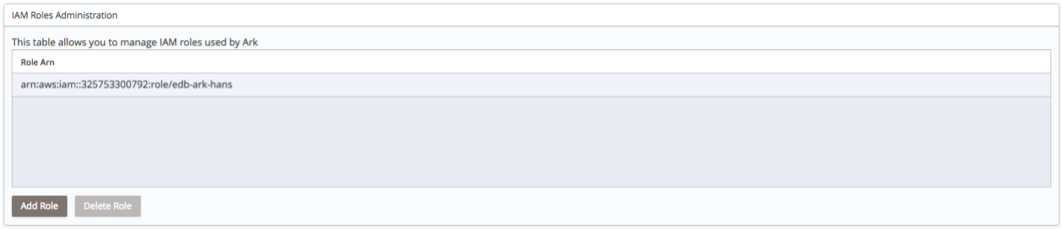


Fig. 8.29: The Roles Administration dialog

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.

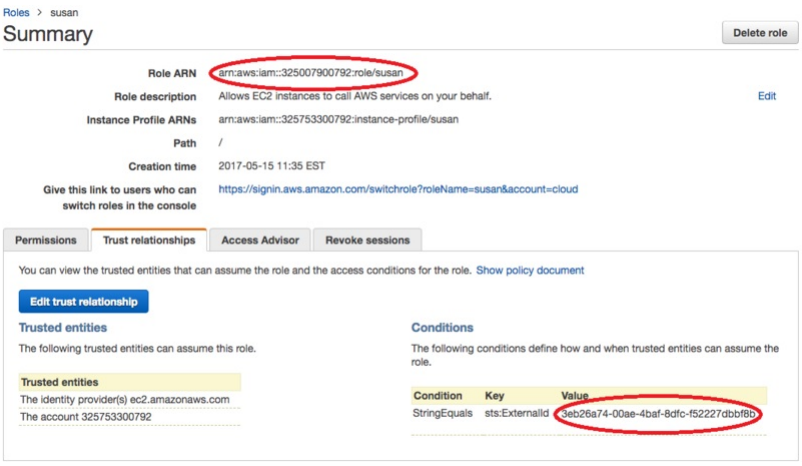


Fig. 8.30: The Roles Administration dialog

To add a Role ARN to the table, click the `Add Role` button; the `Add Role` dialog opens.

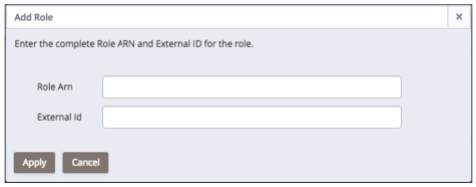


Fig. 8.31: The Roles Administration dialog

Use fields on the 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.

User Administration

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

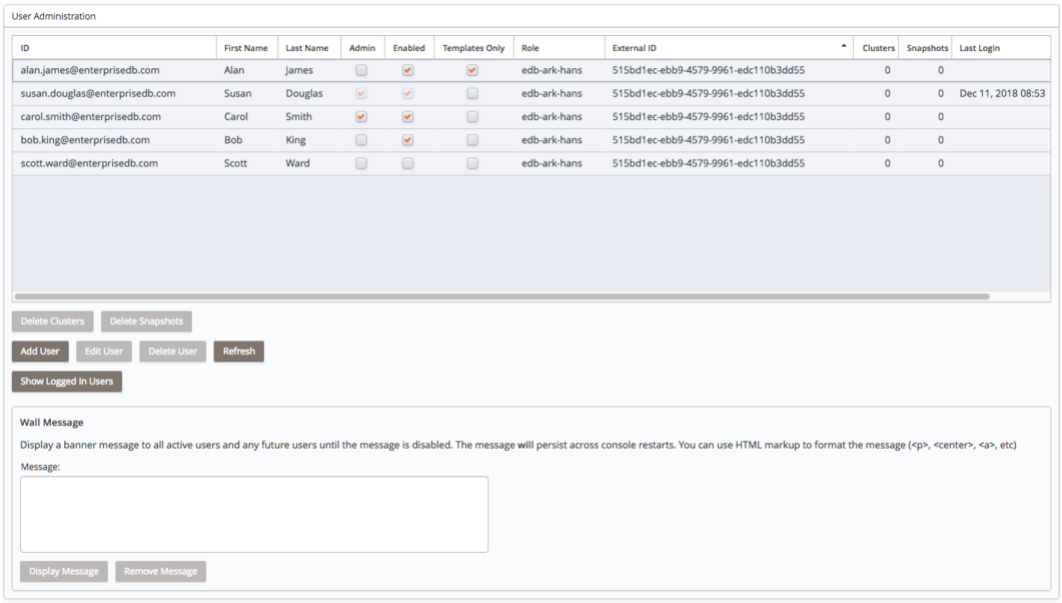


Fig. 8.32: User administration features of the Amazon console

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 and register a new user account for the Ark console.

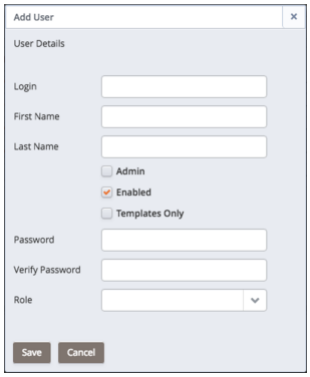


Fig. 8.33: The Add User dialog on an Amazon host

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

User Administration on an Amazon Host

You can use the `User Administration` table to register new users for the Ark console, edit user properties, or delete a user account. 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.

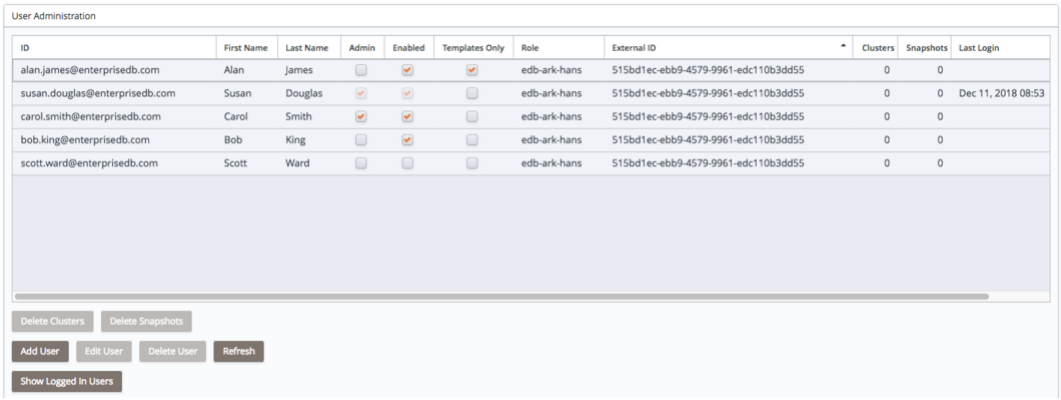


Fig. 8.34: The user table of an AWS console

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.

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.

User Administration on an Azure Host

You can use the `User Administration` table to register new users for the Ark console, edit user properties, or delete a user account. 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.

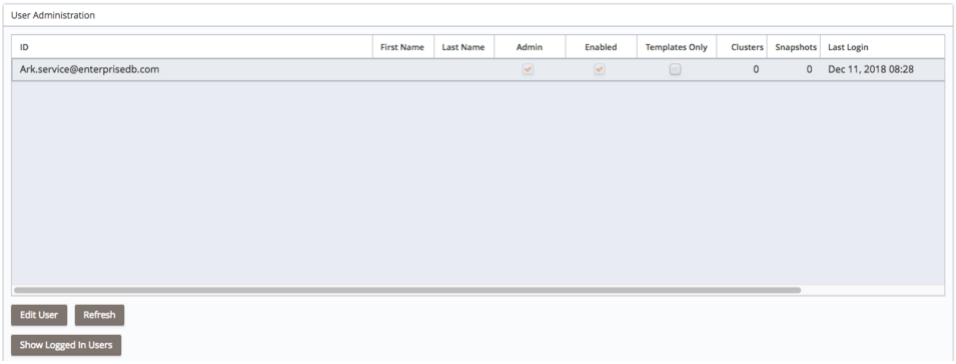


Fig. 8.35: User administration table on the Azure console

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 table.

Displaying Connected Users

Click the `Show logged in users` button to display the `Logged in users` popup.

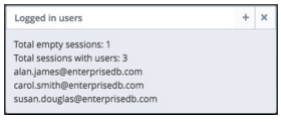


Fig. 8.36: The Logged in users list

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.

Managing the Wall Message

Provide a message in the `Message` field 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.

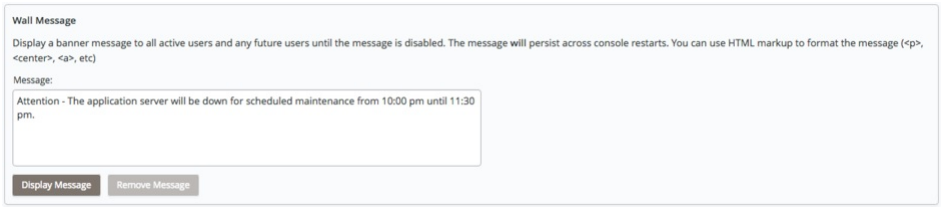


Fig. 8.37: Modifying the Wall Message

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.

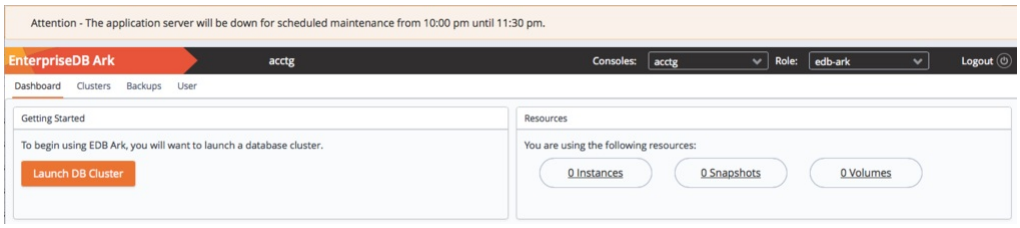


Fig. 8.38: Displaying a wall message

Use the `Remove Message` button to remove the wall banner.

Accessing the Console Logs

Use the `Download Console Logs` panel 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.

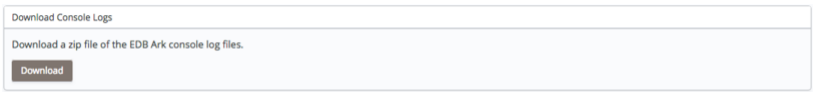


Fig. 8.39: The Download Console Logs section of the Admin tab

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
```

Accessing the Core Usage Logs

Use the `Download Core Usage Logs` panel to download a zip file that contains the core usage by managed clusters on an hourly basis.

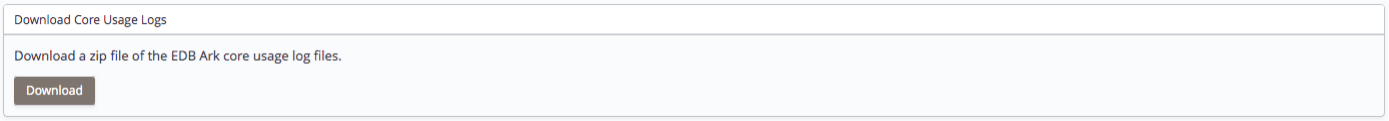


Fig. 8.40: The Download Core Usage Logs section of the Admin tab

The zip file contains a report with four columns; the columns contain:

- The date and time that the entry was written.
- The database type; either `ppas` for Advanced Server, or `postgres` for PostgreSQL.
- The database version.
- The total number of cores running in that region.

Taking a Manual Backup of the Console

Use the `Backup Ark Console` panel to request a manual backup of the Ark console begin.

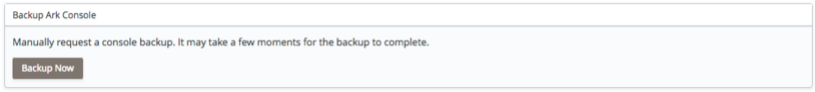


Fig. 8.41: The Backup Ark Console section of the Admin tab

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

Editing Installation Properties

Use the option displayed in the `Edit Installation Properties` panel to review or modify Ark console properties.

Fig. 8.42: The Edit Installation Properties section

Click the `Edit installation properties` button to open the `Edit Installation Properties` dialog. Use fields on the 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, review the AWS deployment instructions.
- For an Azure-hosted console, review the Azure console deployment instructions.

1.9 Using the Ark DBA Tab

The DBA tab displays views that contain information about current clusters and cluster creation history. The tab is accessible only to administrative users.

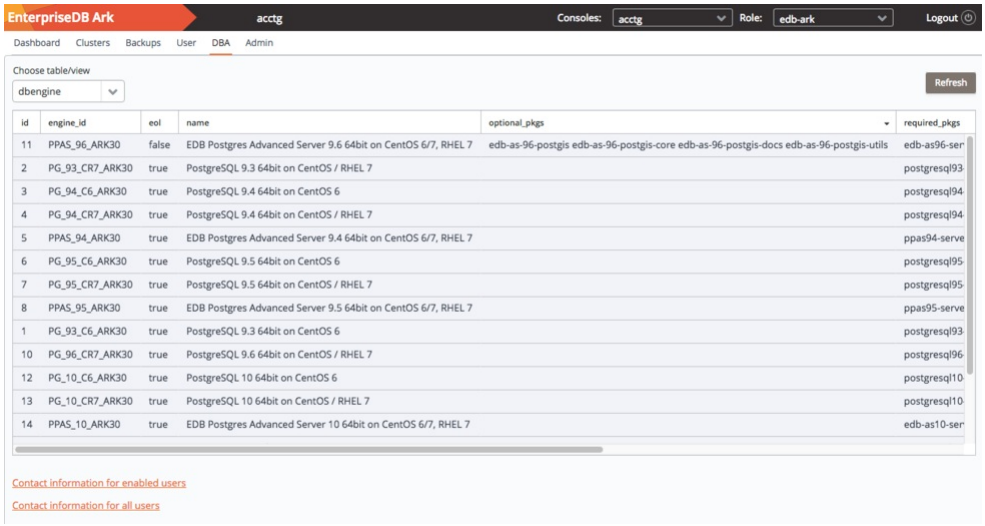


Fig. 9.1: The Ark DBA tab

Use the `Choose table/view` drop down listbox to select a view.

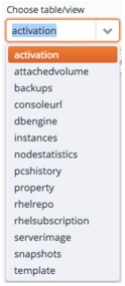


Fig. 9.2: The table/view listbox

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

Fig. 9.3: 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).

1.10 Console Management

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

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
```

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.

Fig. 10.1: Changing console passwords

Fields near the bottom of the dialog allow you to modify the password:

- 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. When prompted, 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.

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`

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/images`

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

The html Template File

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

`/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.

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.

Property Value	Information Logged
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
```

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://<username>:<username>@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 `<username>` and `<username>` 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@ip-172-31-15-135 yum.repos.d]# /usr/share/tomcat/postInstall.sh
Script will upgrade the application! Is the EDB-ARK console in a steady state (no
logged in users, no activity in the console)?
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!
```

When the `yum update` completes, navigate to the Ark URL in your browser, and enter the account password. You will be taken to the Ark deployment dialog where you will be required to provide the `Service Account ID` and `Service Account Password`; provide any missing values, select a `Time Zone`, and restart the Ark server. Ark will open to the `Login` dialog.

Updating a PEM Installation

If you are using a local installation of PEM to monitor an Ark console, you may want to update your version of PEM 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 - $basearch
baseurl=https://<username>:<password>@yum.enterprisedb.com/tools/redhat/rhel-$releas

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.com/depe

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 the EnterpriseDB website.

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 install edb-pem-python-idna.noarch
   ```

   ```
   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.

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.

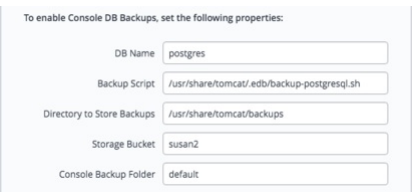
If the system hosting the application database fails, then all information about the console database and registered users will be lost unless you have retained a backup.

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.

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.



When the `Edit Installation Properties` dialog opens, you can modify details about the console backup storage:

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

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.

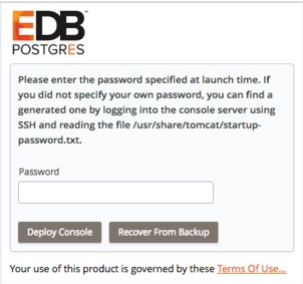


Fig. 10.2: The connection dialog

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.

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.

1.11 Securing EDB Postgres 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.

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.

Fig. 11.1: Editing the Inbound Rules

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.

Consult the core documentation for more information about specifying a CIDR address.

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.

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.

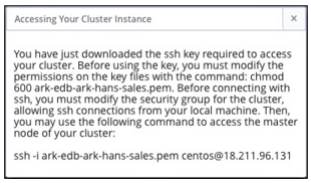


Fig. 11.2: Accessing your Cluster Instance

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
```

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.

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-created user should lose ssh access for any reason.

1.12 Creating a Statically Provisioned Image

An `install.sh` script is distributed with Ark; use the script when creating a statically provisioned image. 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.rpm -p
postgresql10-server pgpool-II-10
```

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.

1.13 Ark 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.

Subject	Body
Continuous Archiving State Changed to *db_state*	A problem was detected with continuous archiving on the master/replica database server *dns_name* in cluster *cluster_name*.
Data Storage Scaling *cluster_name*	Data storage is being added to cluster *cluster_name* because the auto-scaling threshold was reached.
Data storage scaling for cluster *cluster_name* has been suspended	Data storage scaling for cluster *cluster_name* has been suspended. Instance *instance_id* no assignable device names left
Rebuild of primary node in cluster *cluster_name*	The primary server, node id *instance_id* in cluster *cluster_name* is being rebuilt.
Replacement of primary node in cluster *cluster_name*	The primary server, node id *instance_id* in cluster *cluster_name* is being replaced with node id *instance_id*.
Rebuild of replica node in cluster *cluster_name*	The replica server, node id *instance_id* in cluster *cluster_name* is being rebuilt.
Replica promotion failed in cluster *cluster_name*	Replica promotion failed. Performing rebuild of primary DB node; id: *instance_id*
Replica promotion failed in cluster *cluster_name*	Replica promotion failed. Node id: *instance_id*
WARNING: Connectivity Issue with instance *region/instance_id*	WARNING: The EDB Ark cluster manager was unable to connect to the node manager for instance ID *region/instance_id*. This may be due to a temporary connectivity issue or the instance may require manual intervention.
(PITR) Base Backup of cluster *cluster_name* failed	The automatic|manual backup of cluster *cluster_name* in location *availability_zone* failed.
Backup of cluster *cluster_name* failed	The automatic|manual backup of cluster *cluster_name* in location *availability_zone* failed.
WAL Archive Storage 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.

Subject	Body
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 clusters 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.

1.14 Ark Resources

You can 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: 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 is always available from the `Dashboard` tab of the Ark console.

1.15 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",
        "ec2:ImportKeyPair"
      ],
      "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*", "s3 DeleteBucket*"],
      "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"
    }
  ]
}
```

1.16 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"
        }
      }
    }
  ]
}
```

1.17 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": [
                "*"
            ]
        }
    ]
}
```

2 Ark

EDB Postgres Ark

EDB Ark automatically provisions PostgreSQL or EDB Postgres Advanced Server databases in single instances, high-availability clusters, or application development sandboxes. EDB Ark frees DBAs and application developers from the rigors of setting up and administering robust database environments. In minutes, EDB Ark configures a cluster of database machines with:

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

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

This document will demonstrate how to use the EDB Ark console successfully in your cloud-based database management activities.

2.1 What's New

The following features have been added to the EDB Ark user console for this release.

- Ark now supports EDB Postgres Advanced Server and PostgreSQL version 12 database clusters.
- EnterpriseDB has introduced a new RPM repository structure that combines the contents of the previous three repositories into a single repository. The new structure is supported by Ark. For information about connecting to the repository, visit https://info.enterprisedb.com/rs/069-ALB-339/images/Repository%20Access%2004-09-2019.pdf.
- Ark now provides extended support for multi-region features (including manual promotion and API support).
- You can use the `Promote` option (located on a cluster's context menu in the `Details` panel) to replace the master node with a standby node. For more information, see the EDB Postgres Ark Getting Started Guide.
- The `/templates` and `/templates/id` resources now support the `region` property.
- Support for Amazon `r4.*` and `r5.8x` instance types has been reinstated.

Limitations

Cloning, Recovering, or Scaling Encrypted Clusters from Previous Versions

Encrypted clusters created with Ark 3.3 or prior may not be used to clone, recover, or scale to a machine type that is not supported by that earlier version. If you will be moving encrypted clusters created with Ark version 3.3 or prior to a new

machine type (as supported by Ark 3.4 or 3.5), you will need to:

1. Clone the encrypted cluster to a new, unencrypted cluster.
2. Upgrade the Ark console on which the cluster resides.
3. Clone the unencrypted cluster to a new encrypted cluster of the new machine type.

For detailed information about cloning a cluster, see the *EDB Postgres Ark Getting Started Guide*.

Cloning with a Template from a Foreign Region

You cannot use a template when cloning from a foreign region.

Monitoring a Federated Console with PEM

Ark does not support federated consoles configured in local PEM server mode.

Federating Consoles that host clusters with the Same Name

If you are upgrading to version 3.4 or 3.5, and plan to federate existing Ark version 3.3 consoles that host clusters with the same name, you must first create a clone of one of the clusters specifying an alternate name; federated consoles cannot be used to host clusters with the same name. For example, if you plan to federate two consoles that both contain a cluster named `acctg`, you should clone one of the clusters specifying an alternate name for the cluster (i.e. `acctg-west`). Then, you can federate the consoles; both consoles will have access to `acctg` and `acctg-west`.

2.2 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 Advanced Server, EDB Ark also provides a platform with compatibility with the Oracle database, offering dramatic cost savings and competitive advantages.

A cloud is a collection of virtual machines; each virtual machine runs a separate copy of an operating system and an installation of Postgres.

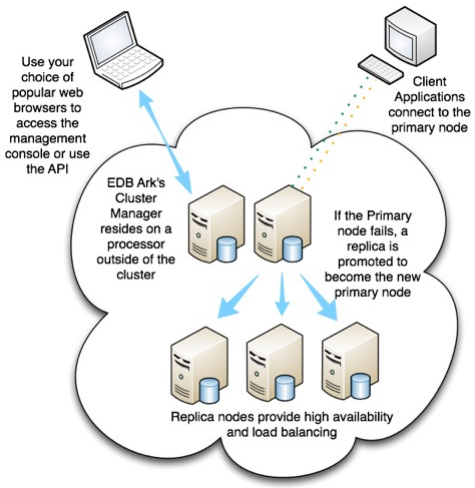


Fig. 2.1: Using EDB Ark in a Private Cloud.

You can specify different combinations of CPU speed, RAM, and disk space to suit your needs when provisioning an EDB Ark cluster. EDB Ark makes it easy to scale up to a more capable cluster, or scale down as your requirements change.

EDB Ark solves common challenges faced by businesses that need more agility, velocity, and thrift in deploying and using relational, ACID-compliant databases:

- Develop / Test / Deploy. Quickly create and delete Postgres databases with standard configurations to support software development and testing activities, then deploy applications to the database or cluster – all at a pace dramatically quicker than physical provisioning.
- Workload Portability. The same Postgres database trusted in the datacenter also runs in a cloud cluster with scalability and high-availability.
- Enterprise-class power. Postgres was designed to solve critical business challenges requiring reliable, high-performance, ACID-compliant database processing. As the only open source database meeting those requirements, it offers an extremely attractive alternative to more expensive options.

EDB Ark includes the following functionality:

- Scale computing resources up and out. EDB Ark automatically scales up storage capacity, and provides a simple button to scale your server class up when data processing loads and usage characteristics require a change in the underlying virtual machine resources.
- Automatic Connection Pooling and Load Balancing. EDB Ark maintains a cluster of database nodes, automatically scaling out replicas based on increasing user demand. The integrated connection pooling manager increases database read performance by distributing requests across all cluster members.
- Self-Healing Failover. EDB Ark automatically replaces downed database nodes, preserving the continuity and performance of the cluster. Users can choose to replace the master with a new master (preserving all committed transactions) or with a promoted replica (for faster recovery time).
- Automatic Online backup. EDB Ark uses user-directed rotating backups to protect your data from loss due to mishaps
- Supports data encryption. EDB Ark offers SSL data encryption that protects data at rest, and is transparent to connecting clients.
- Cost-saving Compatibility with the Oracle Database. Using a database that is compatible with Oracle is a reliable, fast and cost-effective way to support Oracle applications in public and private clouds.
- Web-based interface. EDB Ark provides easy to use point-and-click cluster lifecycle management from start to finish from your favorite web browser.
- Database Cloning. EDB Ark allows you to quickly and easily create developer 'sandboxes' based on real production data, saving System Administrators setup, configuration and data load time.
- Professional Postgres Support. EnterpriseDB provides support from Postgres experts who work with top Postgres open source developers.
- JSON Compatible API Support. EDB Ark supports a JSON-compatible API.

Architecture Overview

The Ark console is designed to help you easily create and manage high-availability database clusters from a web browser.

!!! Note 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. The following figure provides a general overview of the EDB Ark architecture.

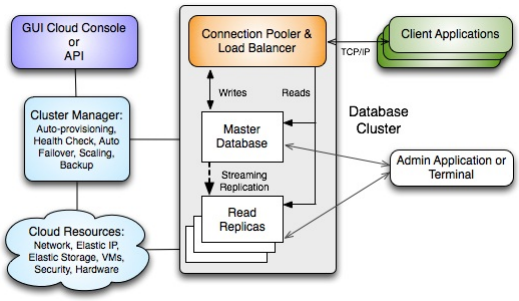


Fig. 2.2: An overview of the EDB Ark architecture

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. 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.

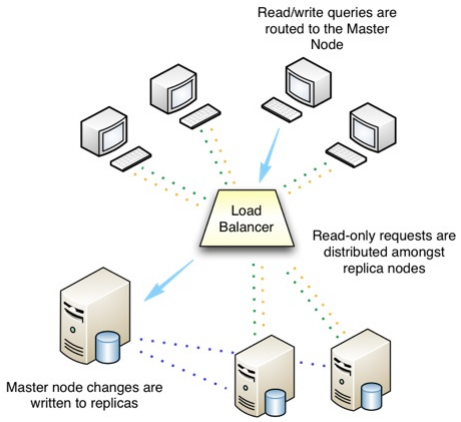


Fig. 2.3: EDB Ark performs automatic load balancing

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; 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.

Using Ark as a Template Only User

Some features of the Ark console are not available to a *Template Only* user. A Template Only user:

- must specify a template when deploying, scaling, or restoring a cluster.
- is restricted to the scaling policies defined in the template.
- cannot modify a manually-defined cluster created by another user.
- can only scale clusters to a server class that exists in a template that is available to the current tenant.
- may only delete backups of template created clusters.
- may not delete last backup of a template created cluster if the cluster had been deleted.

If you are a Template Only user, the Ark console displays a note in the upper-left header when you log in.

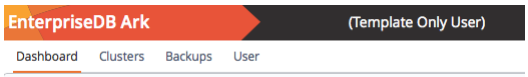


Fig. 2.4: Template Only User

For Template Only users, the Ark dialogs used to create a cluster, clone a cluster, or to restore a backup offer a subset of the fields presented on the dialogs viewed by a user that is not a Template Only user.

For detailed information about using a template to:

- create a cluster, see Using a Template to Create a Cluster .
- restore from backup, see Using a Template to Restore from Backup .
- clone a cluster, see Using a Template to Clone a Cluster .

Please note: a user that is not restricted to template usage may override template policy when modifying a cluster created with a template.

Using a Federated Console

An Administrative user can create a federation of consoles; once federated, a user can create or clone clusters or add resources in any region in which one of the federated consoles resides. Standby nodes that reside in other regions will reflect the state of the master node.

Standby nodes that reside in a different region than the master console will be used for load balancing. Failover to nodes that reside on a different region than the master node is not supported. If a master node fails and you do not have a standby node in the same region as the master node, Ark will create a replacement node in the same region as the original master.

If a console is a member of a federation, any other member of the federation will be able to create resources in the region in which the console resides. All of the resources that reside on federated consoles will be visible on the `Clusters` tab of all of the consoles within the federation.

When connected to a federated console, you can:

- Create a cluster with nodes that reside in multiple regions; the console from which you create the cluster will be responsible for managing the master node.
- Initiate a cluster backup on any cluster within the federation. If the cluster master is managed by the console where the backup was initiated, then the backup will run locally; if the cluster is not managed by the console where the backup was requested, then a backup request will be made to the managing console and the backup will run remotely.
- Delete a node or cluster regardless of the region in which the node or cluster resides.

- Initiate a clone operation regardless of the region in which the node resides; the clone operation will be able to create one or more nodes in any region of the federation.
- Initiate a yum update on any node or cluster of the federation (remote or local).
- Add resources and replicas in any region of a federated console during a manual scaling operation.
- Perform multi-region machine scaling. If the cluster contains nodes running in different regions, then the console that manages the master node will coordinate machine scaling in the foreign regions. A new master node is created first on the new machine type, the old master is removed, and then each replica node is replaced.

On a federated console, dialogs that allow you to create and manage clusters or nodes (create, scale, clone, or delete) will make the regions, VPCs, availability zones, and subnets of other federation members available for selection.

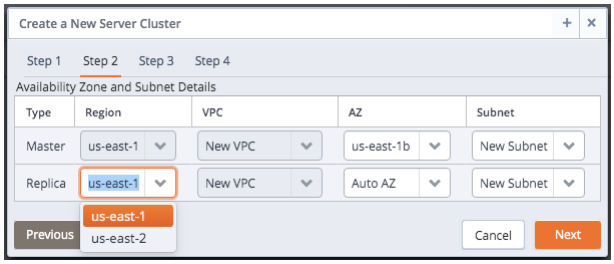


Fig. 2.5: Selecting a region

For example, you might have two federated consoles; one that resides in `us-east-1` and one that resides in `us-east-2`. After federating the consoles (sharing the URL and token of each console within the federation with the other console), you will be allowed to select the `Region` in which replica nodes are created or cloned.

2.3 Accessing the Ark Console

If your Ark console resides on an Amazon host, the console configuration will determine the user registration process. An administrative user can enable or disable self-registration. If you are an administrative user and need information about enabling or disabling self-registration, please refer to the EDB Ark Administrative User's Guide. If you are a non-administrative user connecting to the Ark console on an Amazon host with self-registration enabled, see Using Self-Registration on an Amazon Hosted Console.

When you navigate to the URL of an installed Ark console, the console will display a login dialog.

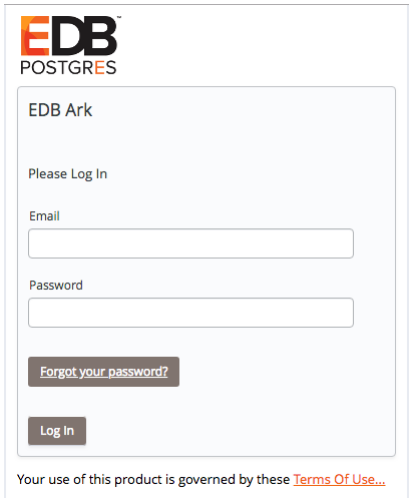


Fig. 3.1: The Login dialog.

Enter your user name in the `Email` field, and the associated password in the `Password` field, and click `Log In` to connect to the Ark console.

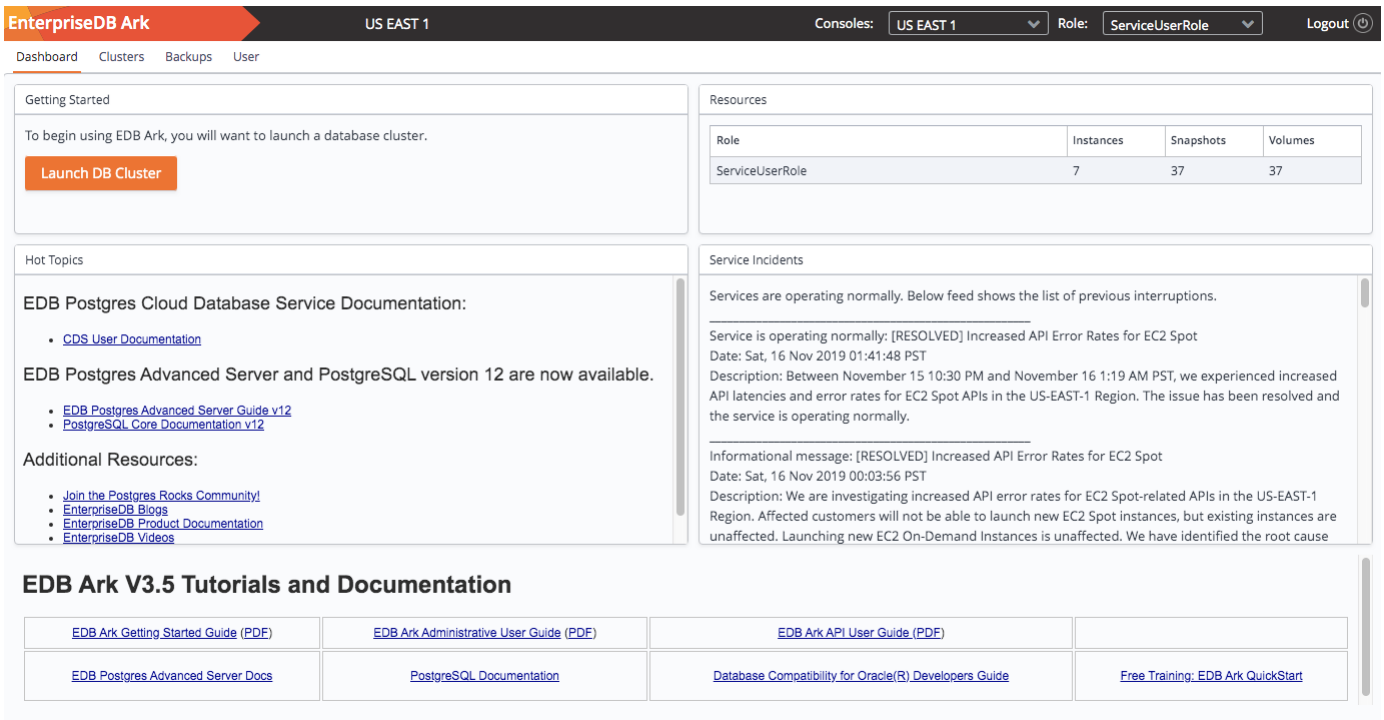


Fig. 3.2: The Ark console.

Using Self-Registration on an Amazon Hosted Console

If self-registration is enabled, on your first visit to the Ark console, you should create an Amazon role and register an Ark console user.

As part of the registration process for the Ark console, you must create an Amazon IAM role and perform a *handshake* between the Ark console and the Amazon management console. The handshake associates the external ID provided by the Ark console with the Amazon role, and the `Role Arn` provided by the Amazon console with the Ark user.

Please note that each time you refresh the Ark `New User` dialog, the external ID displayed on the registration dialog will change; you must have access to both the Ark console and the Amazon management console while registering an Ark user.

To start the registration process, connect to the Amazon management console, and navigate to the `Identity and Access Management` dashboard.

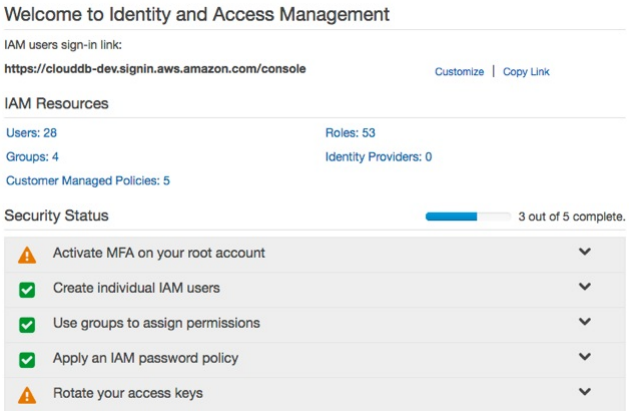


Fig. 3.3: The Amazon IAM Dashboard.

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

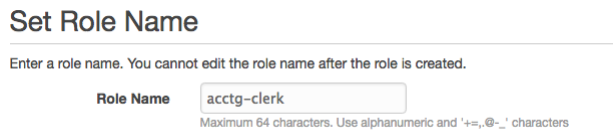


Fig. 3.4: Provide a role name

When the `Set Role Name` dialog opens, specify a name for the new role and click `Next Step` to specify a role type.

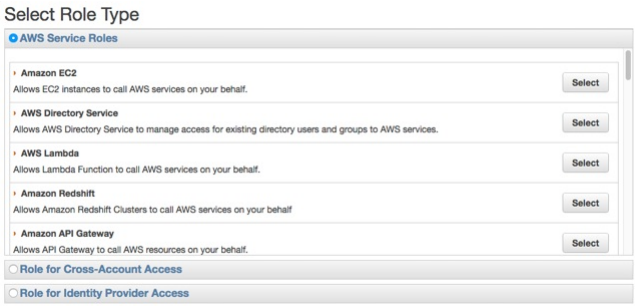


Fig. 3.5: Specify that the role allows EC2 instances to call AWS services

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

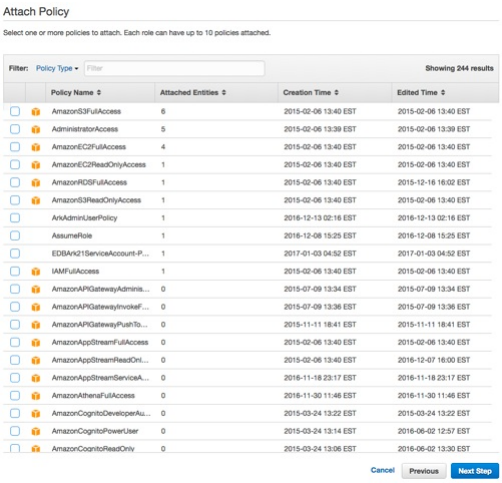


Fig. 3.6: The Attach Policy dialog

When the `Attach Policy` dialog opens, do not specify a policy; instead, click `Next Step` to continue to the `Review` dialog.

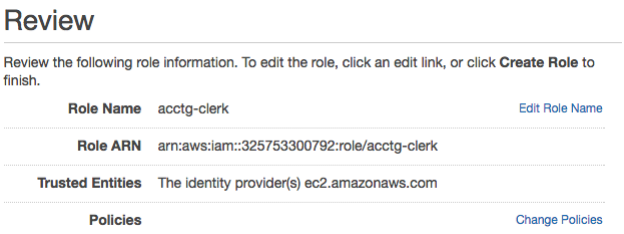


Fig. 3.7: Review the role information

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

Fig. 3.8: The new role is displayed on the Roles page.

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

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

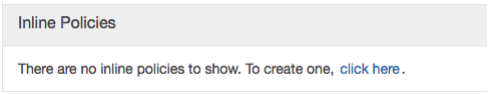


Fig. 3.9: The Inline Policies menu

When the `Set Permissions` dialog opens, select the `Custom Policy` radio button, and then click the `Select` button.

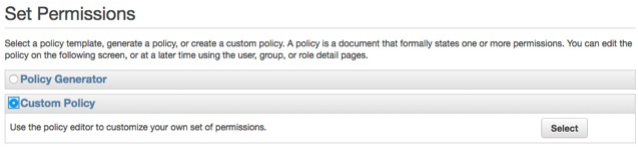


Fig. 3.10: Add a Custom Policy.

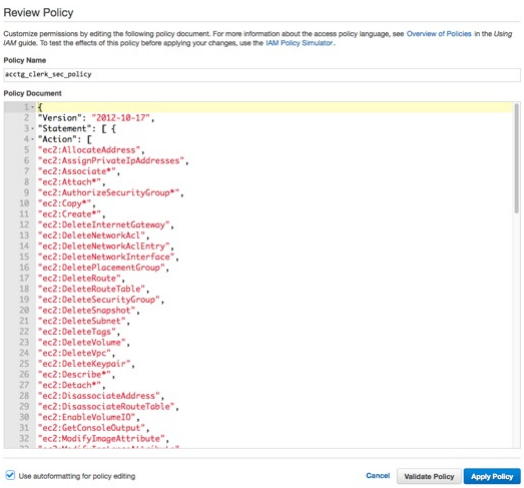


Fig. 3.11: Provide the policy name and contents

Use the fields on the `Set Permissions` dialog to define the security policy:

- Provide a name for the security policy in the `Policy Name` field.
- Copy the security policy text into the `Policy Document` field. The security policy required by Ark is available in AWS Resources .

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

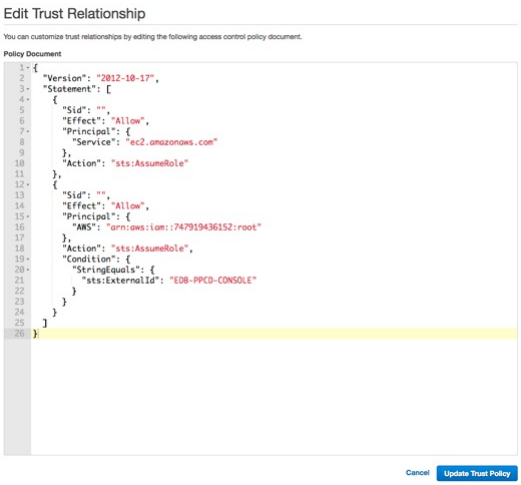


Fig. 3.12: The Policy Document

Replace the displayed content of the policy document with the content of the file available in AWS Resources .

`EDB-PPCD-CONSOLE` 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. Click the `Register` button to open the `New User Registration` dialog.

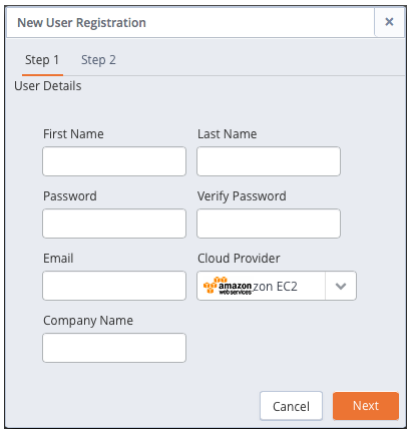


Fig. 3.13: The New User Registration dialog.

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 access the `Step 2` tab.

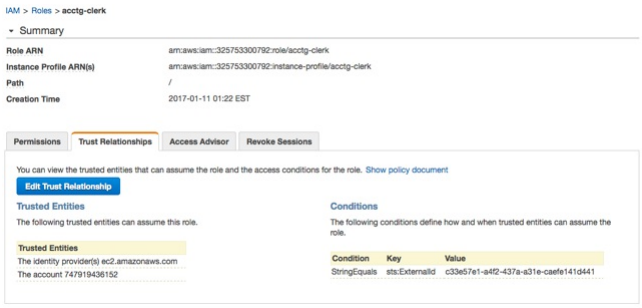


Fig. 3.14: The Summary tab of the Role detail panel.

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

Your Amazon IAM role ARN is displayed on the `IAM Roles` detail panel of the Amazon management console. Highlight a role name to display the assigned value on the `Summary` page.

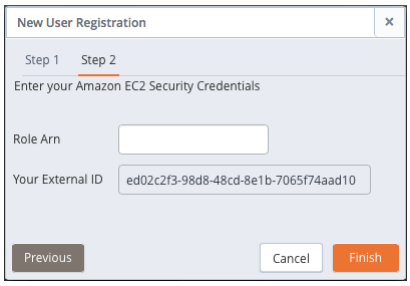


Fig. 3.15: Registering a user on an Amazon EC2 cloud.

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

After completing the registration, you can use the `Login/Register` dialog to access the Ark console.

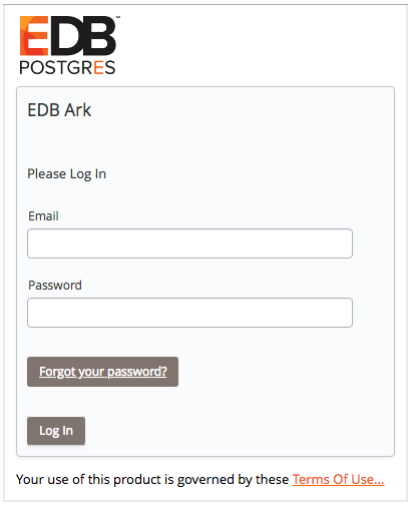


Fig. 3.16: The Login/Register dialog.

Enter the registered email address in the `Username` field, and the associated password in the `Password` field, and click `Log In` to connect to the Ark console.

2.4 Using the Ark Console

The Ark console has four browser tabs that help you to manage clusters that reside in an AWS or Azure cloud:

- The `Dashboard` tab provides an overview of the current resources in use, a quick-start Launch DB Cluster button, and (optionally), links to documentation and tutorials.
- The `Clusters` tab provides a management and information resources for your clusters.
- The `Backups` tab provides a list of the existing snapshots of your Ark clusters and buttons that allow you to create or delete a snapshot.
- The `User` tab provides a graphical management interface that you can use to review and manage your user account information.

Please note that some features of the Ark console are not available to a Template Only user.

2.4.1 The Dashboard Tab

The `Dashboard` tab provides an overview of the EDB Ark service status, resources, useful information links and a quick-start `Launch DB Cluster` button.

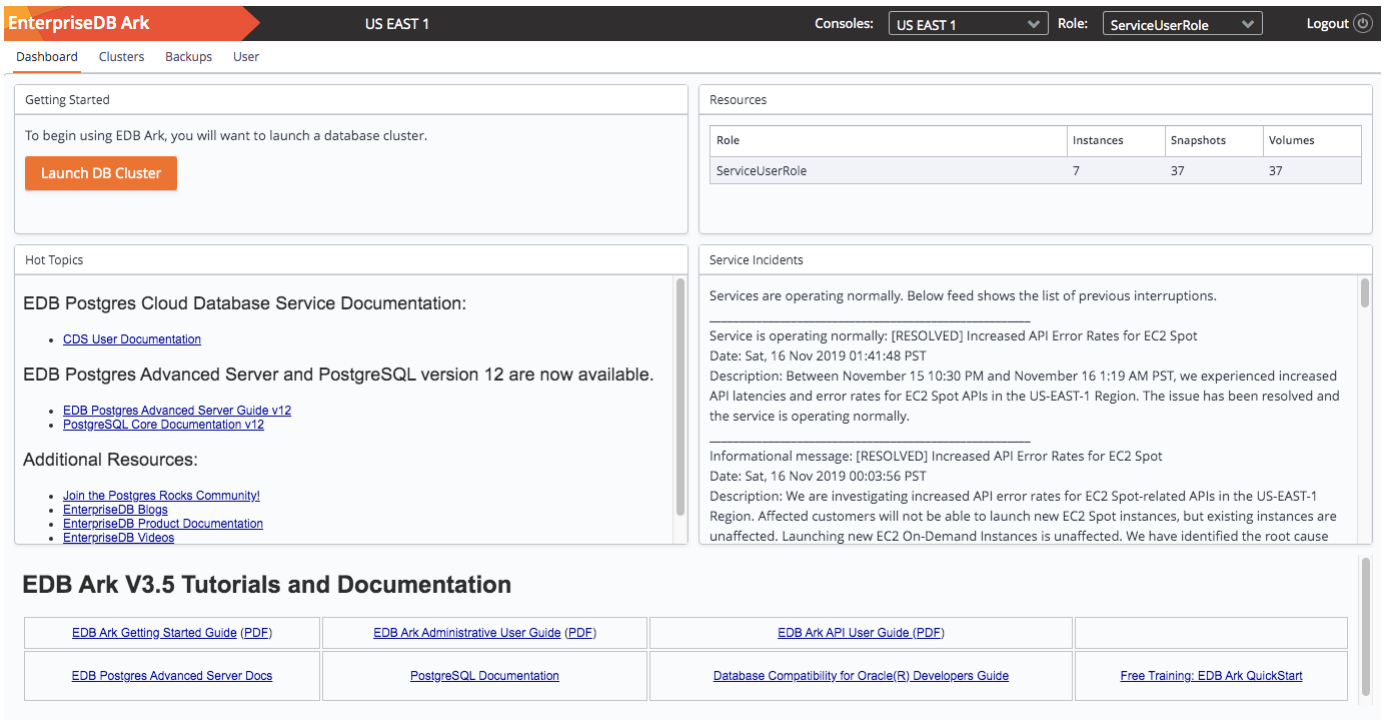


Fig. 4.1: The Dashboard tab.

To launch a cluster from the `Dashboard` tab, use the `Tenant` drop-down listbox to select the tenant in which the cluster will be created. Then, use the `Launch DB Cluster` button located in the `Getting Started` panel to open the `Create New Cluster` dialog and define the cluster attributes. For more information about defining a cluster, see Creating a Server Cluster .

The `Resources` panel contains a table that displays the resource usage (instances, snapshots, and volumes) for each

role.

The `Hot Topics` panel provides a link to the EDB Ark website.

Use the links in the `EDB Ark Tutorials and Documentation` section to access EDB Ark and Postgres documentation.

Using the Console Switcher Feature

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

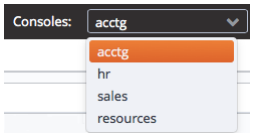


Fig. 4.2: The Consoles drop-down

An Ark administrative user can use management features located on the Admin tab of the administrative console to add consoles to the list, or remove consoles from the list. For more information about populating the console switcher, please see the *EDB Ark Administrative User's Guide*.

2.4.2 The Clusters Tab

Use the `Clusters` tab to create, monitor and manage active clusters that reside in the cluster.

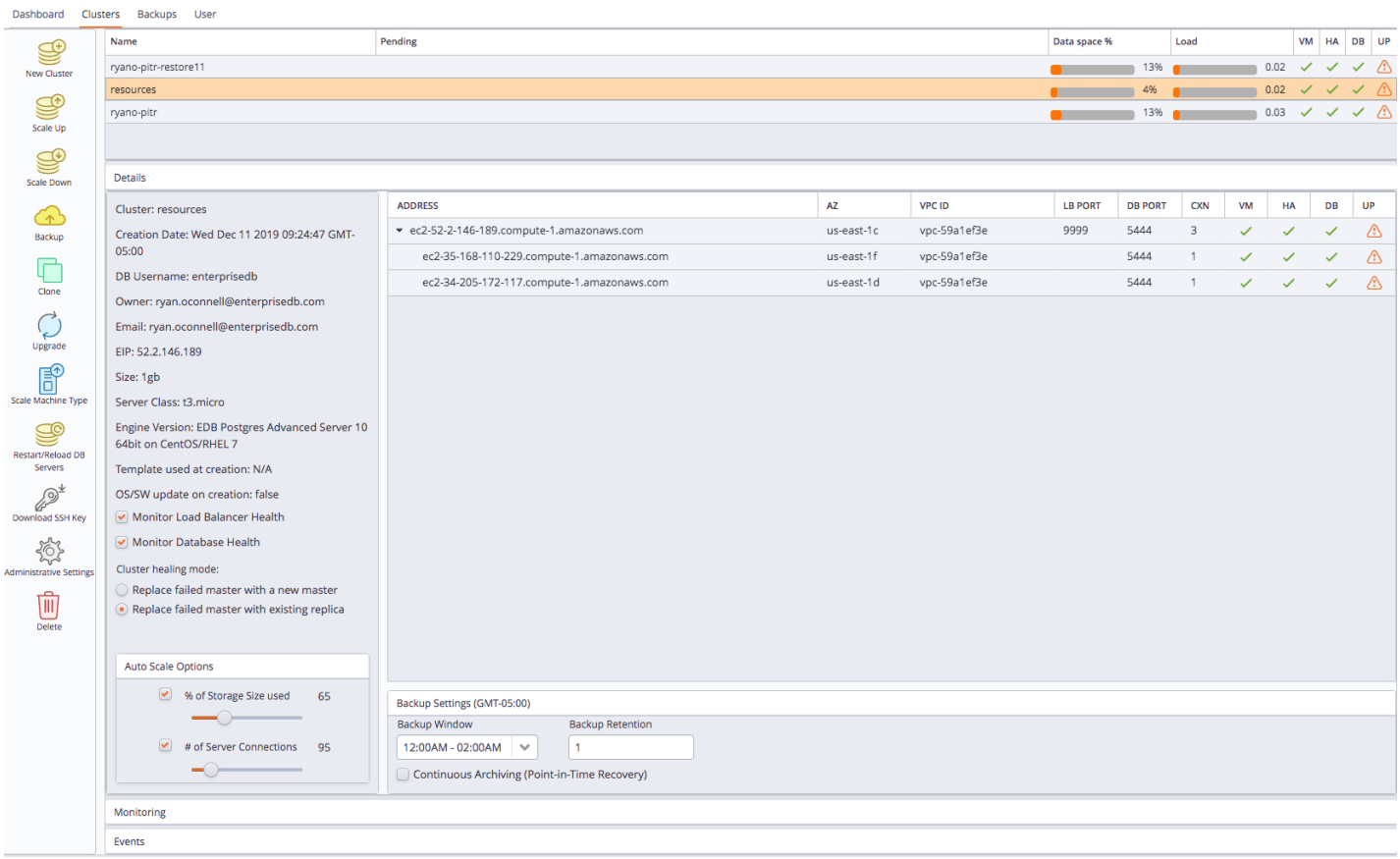


Fig. 4.3: The Clusters tab

Indicators in the columns to the right of a cluster name display the current health of the cluster. Click on a column name to sort the contents of the column; click a second time to reverse the sort-order.

- The `VM` column displays the state of the virtual machine on which the cluster resides.
- The `HA` column displays the state of the high-availability cluster.
- The `DB` column displays the state of the database server.
- The `UP` column displays the current status of the packages installed on the cluster. Periodically, the cluster manager performs a check to see if the packages are up to date.

Status indicators provide quick visual feedback about each feature:

✓	A green checkmark indicates that an object is healthy.
⚠	A yellow alert symbol calls attention to an object that requires attention.
⊗	A red error symbol signifies that an object is not available.
	A busy-indicator signals that the cluster is processing a request.
?	A question mark indicates that the state of the resource is unknown.

Management options are on the node's context menu; right-click the node name to access the menu. The menu is context-sensitive; when applicable, select from:

- `Download DB Logs` to download database server logs.
- `Download pgPool Logs` to download pgPool (load balancing) logs.
- `Promote` to promote the selected standby node to master.

If you choose to download a log file archive, the archive name is in the following form:

```
<address>-<log_type>-<date>.tar.gz
```

Where:

address is the address of the selected node.

log_type is the type of log file (server type or pgPool).

date is the date that the archive was generated.

Use the icons along the left side of the `Clusters` tab to create new clusters or manage existing clusters:

Status indicators provide quick visual feedback about each feature:

Use the `Add Cluster` icon to create a new Postgres cluster.
Select the `Scale Up` icon to manually add one or more replicas to the current cluster, or add additional storage to the current cluster servers.
Use the `Scale Down` icon to remove one or more specified replicas from the cluster.
Select the `Backup` icon to take a backup of the highlighted cluster (a single backup of the cluster data, and a backup of the cluster configuration files).
Select the `Clone` icon to create an exact duplicate of the master node of the selected database. When you clone a database, only the master node is recreated in the new cluster.
Use the `Upgrade` icon to perform a yum update (keeping the same server version) or perform an upgrade to a later server version. After the update, the cluster nodes will be rebooted (initiating any kernel updates required). Please note that a software update can take some time to complete.
Use the `Scale Machine Type` icon to change the size of the virtual machine for the selected cluster. EDB Ark will copy the cluster into a new cluster of a different server class (RAM and CPU), and optionally re-assign the IP address of the existing cluster to the new cluster.
Use the `Download SSH Key` icon to download the SSH key associated with the selected cluster. Each cluster has a unique key that you can use to access nodes in that cluster. Before downloading the SSH key, you should disable pop-up blocker software from restricting pop-ups from the URL/s used by the Ark console or Ark clusters.
Use the `Restart/Reload DB Servers` icon to reload one or more nodes of a selected cluster. Select the instance you wish to restart or reload on the `Restart/Reload DB Servers` dialog and specify if you would like to `Restart` or `Reload` the server before clicking `Confirm` to initiate the action.
Use the `Cluster Administrative Settings` icon to access a popup dialog that allows you to view or modify the ownership and notification email address of a cluster.
Use the `Delete Cluster` icon to delete the currently selected cluster. A popup dialog will ask you to confirm you wish to terminate the cluster; once terminated, a cluster may only be restored from a backup. By default, the box next to `Release elastic IP address` is checked. Deselect this option if you wish to retain the IP address for re-use with other clusters. If you release the IP address, it will be made available for use by other clusters. When you terminate an active cluster, backups are not deleted. Backups (including user data) are retained until they are selected and deleted from the Backups tab.

- Select the Details bar to view information about the state of the selected cluster.
- Select the Monitoring bar to view usage statistics for the selected cluster.
- Select the Events bar to review event logs describing activities on the selected cluster.

The Details Panel

Click the `Details` navigation bar to open the `Details` panel.

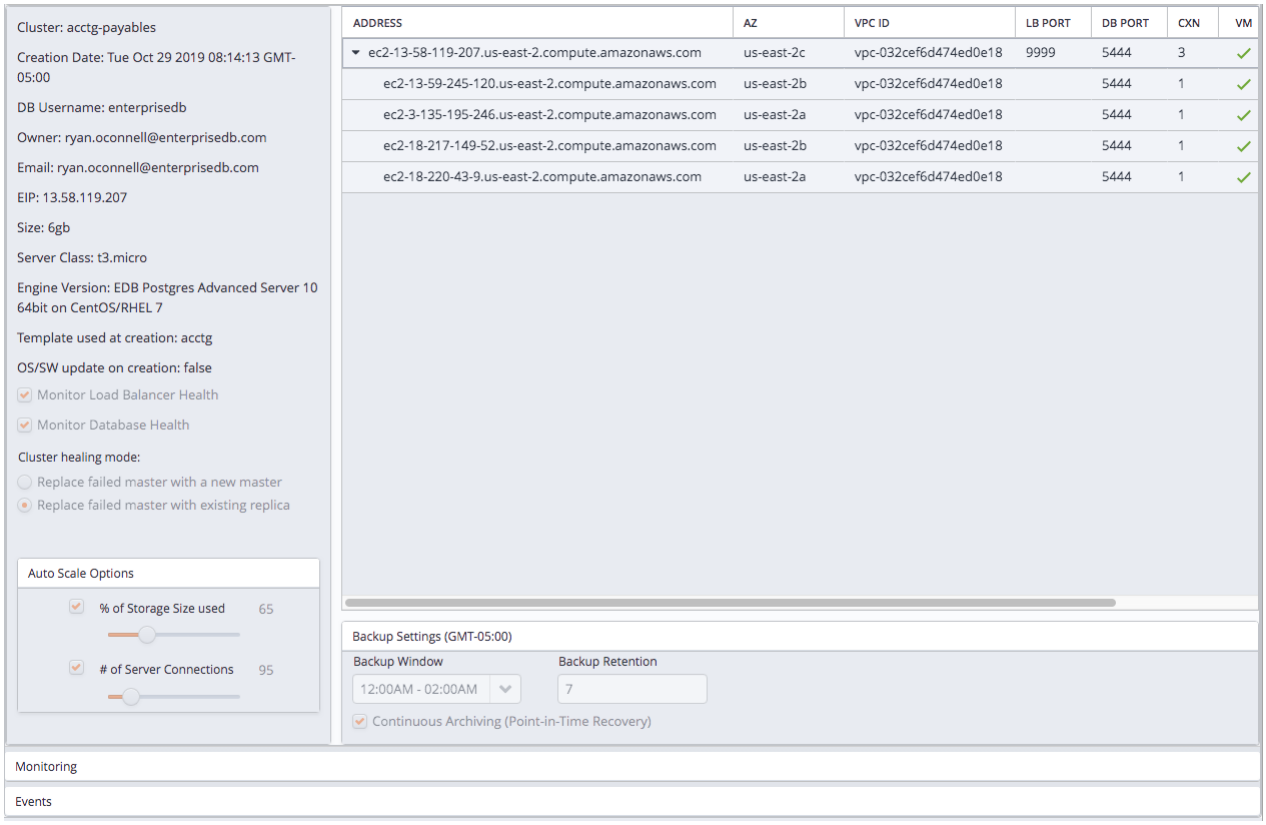


Fig. 4.4: The Details panel on the Clusters tab.

The left pane of the `Details` panel displays information about the currently selected cluster:

- The name of the cluster
- The date and time that the cluster was created
- The name of the database superuser for the cluster
- The name of the cluster owner
- The email address to which notifications about the cluster will be sent
- The elastic IP address of the master node of the cluster
- The cluster size
- If the cluster is encrypted
- If applicable, the IOPS value for the cluster
- The region in which the cluster resides
- The virtual network or VPC ID in which the cluster resides
- The cluster's hardware type or Server Class
- The engine type and version that resides on the server
- If a template was used, the template name
- If the cluster is configured to update when provisioned

You can use controls on the `Details` panel to specify:

- If load balancer health should be monitored
- If database health should be monitored
- Failover preferences for the cluster
- Auto-scaling thresholds for the cluster
- Backup preferences for the cluster
- If continuous archiving should be enabled for the cluster

When you modify the settings on the `Details` panel, EDB Ark displays a `New value saved` notice, confirming that the change has been saved.

Please note: If a template was used to specify the configuration details for the cluster, the template may prohibit access to auto-scaling or manual scaling functionality.

Monitoring Load Balancer Health

By default, EDB Ark monitors the health of the load balancer to ensure that service is not interrupted. If the load balancer (pgpool) should fail while monitoring is enabled, PgPool will be automatically restarted. If the load balancer cannot be automatically restarted, EDB Ark will display a warning sign in the DB column next to the cluster name, and send a notification email to the cluster user.

Deselect the `Monitor Load Balancer Health` checkbox to indicate that you do not wish for load balancer health to be monitored and automatically restarted if an interruption in service is detected.

Monitoring Database Health

By default, the `Monitor Database Health` checkbox is checked, indicating that EDB Ark is monitoring the health of the database to ensure that service is not interrupted. If the state of the database server changes to any state other than running while Monitor Database Health is checked, Ark will attempt to restart the database.

If the database restart fails, Ark will restore the configuration files to their original settings and attempt a restart. If the server fails to restart after restoring the configuration, Ark will failover to a new instance.

Uncheck the `Monitor Database Health` checkbox to instruct Ark to not automatically restart the database if the database stops.

Selecting a Cluster Healing Mode

Use the `Cluster healing mode` radio buttons to specify the type of failover that should be employed:

- Select the `Replace failed master` with a new master radio button to specify that the cluster manager should create a new master to replace a failed master node.

  When replacing a failed master node with a new master node, the data volumes from the failed instance are attached to the new master node, preserving data integrity, while the replicas continue serving client queries.

- Select the `Replace failed master with existing replica` radio button to specify that the cluster manager should promote a replica node to be the new master node for the cluster.

  When replacing a failed master node with an existing replica, a replica node is marked for promotion to master node, while the other replica nodes are re-configured to replicate data from the new master node. Since replica nodes use asynchronous replication, any data that was committed to the old master node, but not pushed to the replica prior to the node failure will be lost.

Please note that replacing a failed master node with a new master node can take a bit longer than promoting a replica node to the role of master, but it does have the advantage of guaranteeing that no committed data will be lost. If recovery time for your cluster is more important than preserving any non-replicated transactions, then select Replace failed master with existing replica as the healing mode.

Adjusting Auto-Scaling Thresholds

Use the `Auto-Scaling Thresholds` controls on the `Details` panel to adjust the threshold at which EDB Ark automatically scales up cluster resources. For more information about using the controls, see Adjusting the Automatic Scaling Thresholds .

Modifying Backup Settings

Use the fields in the `Backup Settings` box to change your backup preferences for the selected cluster:

- Use the `Backup Window` drop-down listbox to select an optimal time to process cluster backups; specify a time when the number of clients accessing the database is minimal.
- Use the `Backup Retention` field to specify the number of backups that should be stored for the selected cluster.
- Select the checkbox next to `Continuous Archiving (Point-in-Time Recovery)` to enable point-in-time recovery for a 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. Note that if you deselect this option, the cluster (and subsequent automatic backups) will be re-configured to not include support for point-in-time recovery.

  When point-in-time recovery is enabled, the value specified in the `Backup Retention` field determines the duration of the point-in-time recovery backup window. For example, if you specify a value of 7, the backup window will be `7` calendar days long. When the backup retention threshold is reached, the oldest base backup is removed, as well as any WAL files required to perform a recovery with that backup.

Reviewing Cluster Connection and Status Information

The `DNSNAME` table (located on the right side of the `Details` panel) contains a status overview and connection information for the selected cluster. If you have created replicas, the secondary server nodes are listed below the master node in the tree control; expand the tree control to view the status of the replication nodes.

Status indicators provide quick visual feedback about each feature:

✓	A green checkmark indicates that an object is healthy.
⚠	A yellow alert symbol calls attention to an object that requires attention.
⊗	A red error symbol signifies that an object is not available.
	A busy-indicator signals that the cluster is processing a request.
?	A question mark indicates that the state of the resource is unknown.

Use the drop-down tab in the upper-right corner of the `DNSNAME` pane to select the columns that will be displayed in the panel:

- The `AZ` column displays the Availability Zone in which the cluster resides.
- The `VPC ID` column displays the identifier of the VPC on which the cluster resides.
- The `LBPORT` column displays the port number to which a client application should connect to utilize load balancing.
- The `DBPORT` column displays the default listener port for the Advanced Server or PostgreSQL server.
- The `CXN` column displays the current number of connections to the node.
- The `VM` column displays the state of the virtual machine on which the cluster resides.
- The `HA` column displays the state of the high-availability cluster.
- The `DB` column displays the state of the database server.
- The `UP` column displays the current status of the packages installed on the cluster. Periodically, the cluster manager performs a check to see if the packages are up to date. If an update becomes available, the `UP` column will display a

yellow alert symbol if the update is non-critical, or a red error symbol if the update is a critical (security) alert.

If alerted to an out-of-date package, you can use the `Upgrade` icon to invoke a yum update to update the package on all of the nodes on your cluster.

The Monitoring Panel

The `Monitoring` panel displays graphs that allow you to review statistical usage information about the amount of storage and the CPU load for the selected cluster.

Fig. 4.5: The Monitoring panel displays usage information

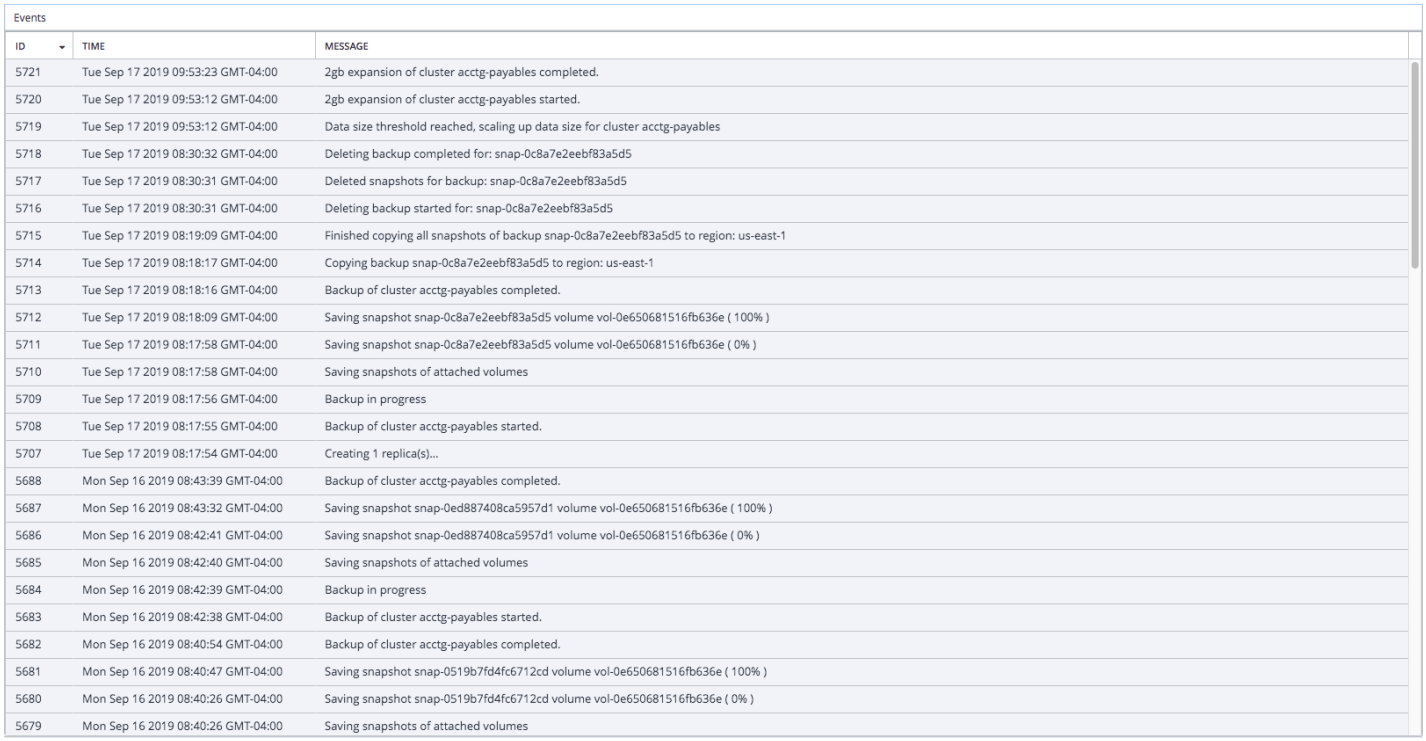
Use the `Time Range` drop-down listbox to modify the time period that the charted information on the Monitoring panel spans.

The graphs on the `Monitoring` panel display resource usage information:

- The `Data Space` chart displays the amount of allocated data space used by the selected cluster. The red line denotes the threshold specified by the `Data Space Threshold` slider on the `Details` panel (the threshold at which the cluster will be scaled-up). The blue line indicates the amount of the data space that is currently in use.
- The `Connections` chart displays a graph of the number of connections to the cluster during the selected time range. The red line denotes the threshold specified by the Connections slider on the Details panel.
- The `Load` chart displays the processing load placed on the CPU by connecting clients. The value displayed is the actual load average as read from the program, `/proc/loadavg`. The chart shows the number of jobs in the run queue or waiting for disk I/O, averaged over 15 minute periods.
- The `Replication Lag` chart displays the replication lag (in seconds) for the cluster. Each replica node is displayed as a separate colored line on the chart; the key at the bottom of the chart identifies the IP address of the node.

The Events Panel

The `Events` panel displays an event log containing a history of selected events for the connected user.



Events

ID	TIME	MESSAGE
5721	Tue Sep 17 2019 09:53:23 GMT-04:00	2gb expansion of cluster acctg-payables completed.
5720	Tue Sep 17 2019 09:53:12 GMT-04:00	2gb expansion of cluster acctg-payables started.
5719	Tue Sep 17 2019 09:53:12 GMT-04:00	Data size threshold reached, scaling up data size for cluster acctg-payables
5718	Tue Sep 17 2019 08:30:32 GMT-04:00	Deleting backup completed for: snap-0c8a7e2eebf83a5d5
5717	Tue Sep 17 2019 08:30:31 GMT-04:00	Deleted snapshots for backup: snap-0c8a7e2eebf83a5d5
5716	Tue Sep 17 2019 08:30:31 GMT-04:00	Deleting backup started for: snap-0c8a7e2eebf83a5d5
5715	Tue Sep 17 2019 08:19:09 GMT-04:00	Finished copying all snapshots of backup snap-0c8a7e2eebf83a5d5 to region: us-east-1
5714	Tue Sep 17 2019 08:18:17 GMT-04:00	Copying backup snap-0c8a7e2eebf83a5d5 to region: us-east-1
5713	Tue Sep 17 2019 08:18:16 GMT-04:00	Backup of cluster acctg-payables completed.
5712	Tue Sep 17 2019 08:18:09 GMT-04:00	Saving snapshot snap-0c8a7e2eebf83a5d5 volume vol-0e650681516fb636e (100%)
5711	Tue Sep 17 2019 08:17:58 GMT-04:00	Saving snapshot snap-0c8a7e2eebf83a5d5 volume vol-0e650681516fb636e (0%)
5710	Tue Sep 17 2019 08:17:58 GMT-04:00	Saving snapshots of attached volumes
5709	Tue Sep 17 2019 08:17:56 GMT-04:00	Backup in progress
5708	Tue Sep 17 2019 08:17:55 GMT-04:00	Backup of cluster acctg-payables started.
5707	Tue Sep 17 2019 08:17:54 GMT-04:00	Creating 1 replica(s)...
5688	Mon Sep 16 2019 08:43:39 GMT-04:00	Backup of cluster acctg-payables completed.
5687	Mon Sep 16 2019 08:43:32 GMT-04:00	Saving snapshot snap-0ed887408ca5957d1 volume vol-0e650681516fb636e (100%)
5686	Mon Sep 16 2019 08:42:41 GMT-04:00	Saving snapshot snap-0ed887408ca5957d1 volume vol-0e650681516fb636e (0%)
5685	Mon Sep 16 2019 08:42:40 GMT-04:00	Saving snapshots of attached volumes
5684	Mon Sep 16 2019 08:42:39 GMT-04:00	Backup in progress
5683	Mon Sep 16 2019 08:42:38 GMT-04:00	Backup of cluster acctg-payables started.
5682	Mon Sep 16 2019 08:40:54 GMT-04:00	Backup of cluster acctg-payables completed.
5681	Mon Sep 16 2019 08:40:47 GMT-04:00	Saving snapshot snap-0519b7fd4fc6712cd volume vol-0e650681516fb636e (100%)
5680	Mon Sep 16 2019 08:40:26 GMT-04:00	Saving snapshot snap-0519b7fd4fc6712cd volume vol-0e650681516fb636e (0%)
5679	Mon Sep 16 2019 08:40:26 GMT-04:00	Saving snapshots of attached volumes

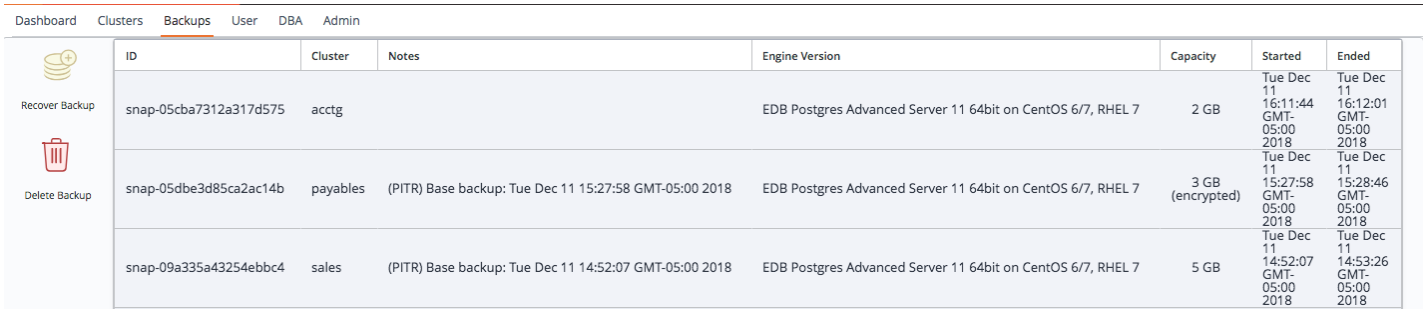
Fig. 4.6: The Events panel displays server activity.

Highlight a cluster name to display only events for that cluster; if you do not select a cluster, the `Events` panel will display the collected events for the connected user.

- Click a column heading to sort the logged activity by the selected column; click again to reverse the sort order.
- Use a mouse to select multiple rows from the event log for copy and paste operations.

2.4.3 The Backups Tab

Use the `Backups` tab to manage cluster backups; the tab displays a list of the available backups.



Dashboard | Clusters | Backups | User | DBA | Admin

Recover Backup

Delete Backup

ID	Cluster	Notes	Engine Version	Capacity	Started	Ended
snap-05cba7312a317d575	acctg		EDB Postgres Advanced Server 11 64bit on CentOS 6/7, RHEL 7	2 GB	Tue Dec 11 16:11:44 GMT-05:00 2018	Tue Dec 11 16:12:01 GMT-05:00 2018
snap-05dbe3d85ca2ac14b	payables	(PITR) Base backup: Tue Dec 11 15:27:58 GMT-05:00 2018	EDB Postgres Advanced Server 11 64bit on CentOS 6/7, RHEL 7	3 GB (encrypted)	Tue Dec 11 15:27:58 GMT-05:00 2018	Tue Dec 11 15:28:46 GMT-05:00 2018
snap-09a335a43254ebbc4	sales	(PITR) Base backup: Tue Dec 11 14:52:07 GMT-05:00 2018	EDB Postgres Advanced Server 11 64bit on CentOS 6/7, RHEL 7	5 GB	Tue Dec 11 14:52:07 GMT-05:00 2018	Tue Dec 11 14:53:26 GMT-05:00 2018

Fig. 4.7: The Backups tab of the Ark console

A backup captures and stores the status and condition of a cluster at a specific point-in-time. Click a column heading to sort the column contents; click again to reverse the sort order.

If the comment in the `NOTES` column for a specific cluster includes `PITR`, point in time recovery is enabled on the cluster. When point in time recovery is enabled, the backup can be used to restore your cluster to a state at any given time since the backup was taken.

Use the icons on the left side of the `Backups` tab to restore or delete backups:

- Highlight a backup in the list, and click the `Recover Backup` icon to open a dialog that allows you to restore a cluster from the selected backup. Specify a name for the cluster, and click the `Recover` button to continue. A popup confirms that the cluster is being restored; navigate to the Clusters tab to monitor the restoration process.
- Highlight one or more backups in the list and click the `Delete Backup` icon to delete the selected backups. A popup will ask you to confirm that you wish to delete the selected backups before the backups are actually deleted.

2.4.4 The User Tab

Fields on the `User` tab allow you to view or modify information about the current user.

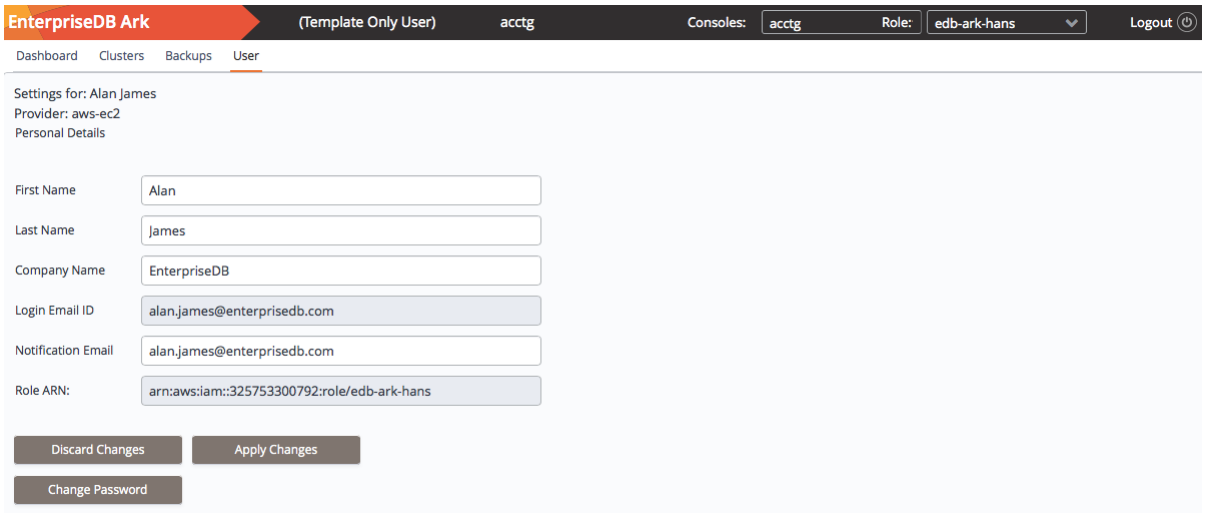


Fig. 4.8: The User tab of the Ark console.

To change the `First Name`, `Last Name`, or `Company Name` of the registered user, modify the corresponding fields and click the `Apply Changes` button. A popup will confirm that the changes have been applied.

The `Notification Email` field on the `User` tab displays the default email address that will be used for cluster notifications unless an alternate address is provided. You can optionally:

- provide an alternate email address when a cluster is created (on the `Create a new Server Cluster` dialog).
- modify a cluster's notification email address on the `Administrative Settings` dialog.

To change the default notification email address, enter a new address in the `Notification Email` field, and click the `Apply Changes` button. A popup dialog will open, prompting you to enter your password to confirm the change of address. Enter your password, and click `Confirm` to modify the address, or click `Cancel` to exit the popup without applying the change.

If you elect to change the notification email address, EDB Ark will send a confirmation email to both the old notification address and the new notification address.

Updating a Password on Amazon AWS

If your Ark console is deployed on Amazon AWS, the `User` tab displays the Amazon Role ARN associated with your Ark user account, and provides an option that allows you to modify your password. To modify your password, click the `Change Password` button.

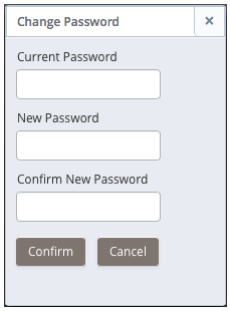


Fig. 4.9: The Change Password dialog.

To modify your password:

- Provide your current password in the `Current Password` field.
- Enter the new password in the `New Password` field.
- Confirm the new password in the `Confirm New Password` field.

Click the `Confirm` button to change the password to the new value; click `Cancel` to exit the dialog without modifying the password.

When you change your password, a popup will confirm that the password has been changed and Ark will send an email to the notification email address associated with the account.

2.5 Creating a Server Cluster

To create a server cluster, you can:

- Use the `Launch DB Cluster` button (located on the `Dashboard` tab) to open the `Create a new Server Cluster` dialog and define a new cluster.
- Click the `New Server` button (located on the `Clusters` tab) to open the `Create a new Server Cluster` dialog and define a new cluster.

If you are not a *Template Only* user (a user required to use a template when defining a cluster), the `Create a New Server Cluster` dialog will prompt you to select a deployment option.

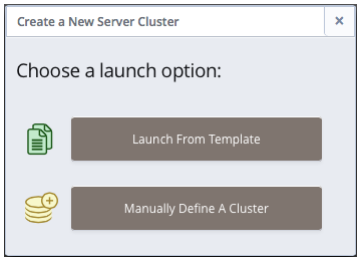


Fig. 5.1: Specify your server launch preferences.

If you are a Template Only user or select the `Launch From Template` option on the deployment method selection

dialog, a dialog opens that allows you to use a pre-defined template for the cluster configuration; for detailed information about using a template to create a cluster, see Using a Template to Create a Cluster .

If you select `Manually Define A Cluster`, a dialog opens that allows you to specify detailed information about the cluster configuration. For information about manually defining a cluster, see Manually Creating a Cluster .

You can also use an existing cluster or a backup as a starting point for a new cluster:

- For information about cloning a new server cluster from an existing cluster, see Cloning a Server Cluster .
- For information about restoring a backup to create a new cluster, see Restoring a Cluster from Backup .

Manually Creating a Cluster

Before you can connect to Postgres from a client application, you must create a server cluster. Use the `Launch DB Instance` button (located in the upper left panel of the `Dashboard` tab) or click the `Add Server` button on the `Clusters` tab to open the `Create a New Server Cluster` dialog.

Please note: not all fields and tabs documented in the following sections are applicable for all console host types.

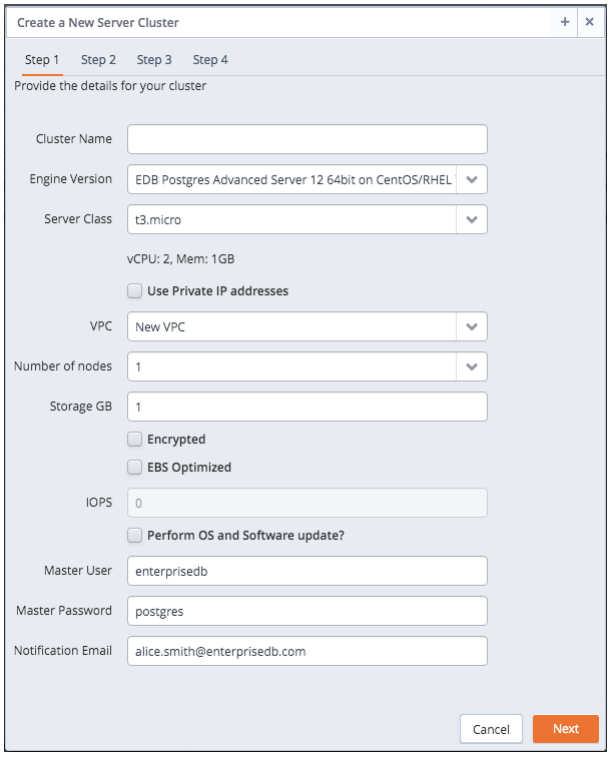


Fig. 5.2: Specify information about the new cluster on the Step 1 tab.

Use fields on the `Create a New Server Cluster` dialog to specify information about the new cluster:

- Specify a name for the new server cluster in the `Cluster Name` field. Ark uses the name specified in the `Cluster Name` field to identify the cluster when performing management functions.
- Use the drop-down listbox in the `Engine Version` field to select the version of the Postgres engine that you wish to use.
- 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 each node within the cluster. When you select a server class, the attributes of the selected class are displayed below the server class field.

You can adjust the amount of storage used by the cluster, or number of replicas in the cluster as your resource demands change. For example, you can start with a m1.small instance, and later, easily upgrade to a more capable c1.medium instance as your performance requirements dictate.

- If your cluster resides on an Amazon host, you can check the box to the left of Use `Private IP` addresses to display addresses that are in your private network in the `VPC` field.
- If your cluster resides on an Amazon host, use the drop-down listbox in the `VPC` field to specify the identity of the network in which the cluster should reside.
- Use the drop-down listbox in the Number of nodes field to specify the number of server nodes that you wish to create. The name specified in the `Cluster Name` field will apply to the master node; each additional node will act as a replication server for the master node.
- 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 your cluster resides on an AWS host, check the box next to `EBS Optimized` to specify that your 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, use the field to 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 instruct EDB Ark to perform a yum update whenever the cluster is provisioned; this option is disabled for clusters that use statically provisioned servers. The yum update command updates all of the outdated packages that reside on the cluster. The update will occur when a cluster is scaled up, scaled down, or during a failover.

  When you check the box next to `Perform OS and Software update ?`, EDB Ark will warn you that enabling this functionality can significantly slow down cluster operations. Updating packages may slow down cluster maintenance operations; an update can easily take 10 minutes or more, and will initiate a reboot of the node. This setting is persistent; if you enable software updates for a cluster, you cannot directly disable software updates for that cluster at a later time.
- Enter the name of the database superuser in the `Master User` field.
- Enter the password associated with the database superuser in the `Master Password` field.
- Use the `Notification Email` field to provide the email address that will receive notices about changes to the cluster status.

If applicable on your host system, click the `Next` button to continue to the `Step 2` tab.

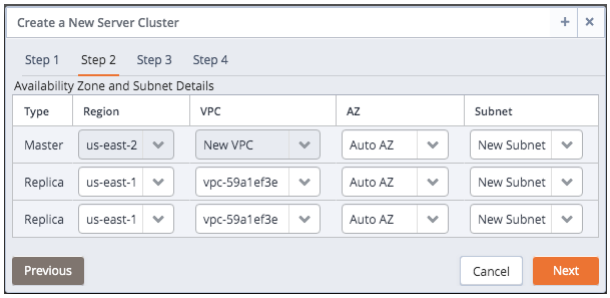


Fig. 5.3: The Step 2 tab.

Use the fields on the `Step 2` tab to specify additional database information for each node of the cluster; the `Type` column identifies if a node is a `Master` or `Replica` node:

- Use the `Region` drop-down listbox to the right of each node to select the region in which the node will reside.
- Use the `VPC` drop-down listbox to select the VPC in which the node will be created.
- Use the `AZ` drop-down listbox to select the Availability Zone in which the node will be created.
- Use the `Subnet` drop-down listbox to the right of each node to select the subnet that the node will use. Please note that if you manually specify a subnet, you must select a subnet that resides on your private network.

Click the `Next` button to continue to the `Step 3` tab.

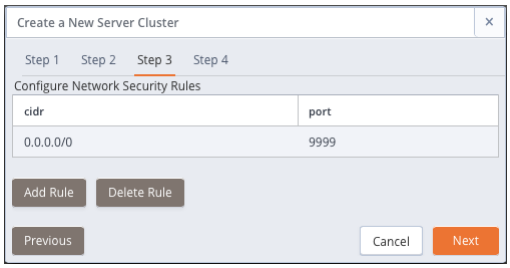


Fig. 5.4: Specify cluster security rules on the Step 3 tab.

Use the fields on the `Step 3` tab to define security rules that allow access to the cluster. By default, the load balancer port is open to any IP address for client connections; you may choose to delete the rule, and specify a more restrictive IP range.

To delete a rule from the list, highlight the entry and click the `Delete Rule` button; you will be prompted to confirm that you wish to delete the entry before the server removes the rule.

Click the `Add Rule` button to open the `Add Rule` dialog and provide access to a port.

Fig. 5.5: Adding a security rule.

On the `Add Rule` dialog:

- Use the `Port` drop-down listbox to select the port that can be accessed from the specified CIDR. A non-administrative user can allow access to ports:

  `9999` - for client connections and load balancing.

  `5432` or `5444` - the cluster specific database listener port.

  An administrative user can use the `Add Rule` dialog to add a rule that allows SSH access to Port `22`.

- Use the `CIDR` field to specify the address (or address range) that will be allowed access to the server through the selected port.

When you're finished, click `Apply` to create the security rule.

Then, click `Next` to continue to the `Step 4` tab.

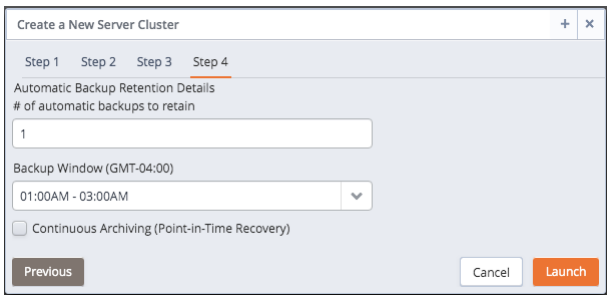


Fig. 5.6: Specify backup information on the Step 4 tab.

Use the fields on the `Step 4` tab to specify additional database information:

- Use the `# of automatic backups to retain` field to specify the number of server backups stored. When the specified number of server backups is reached, EDB Ark will delete the oldest backup to make room for a new backup.

  When point-in-time recovery (PITR) is enabled, the value specified in the # of automatic backups to retain setting determines the duration of the PITR backup window. For example, if you specify a value of 7, the PITR backup window will be 7 calendar days long.

- Use the `Backup Window` field to specify a time that it is convenient to backup the server (you may wish to schedule backups to occur when the CPU load is the lightest).

- 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. Note that if you deselect this option, the cluster (and subsequent automatic backups) will be re-configured to not include support for point-in-time recovery.

  Please Note:

  If your cluster resides on an Amazon host that is running CentOS 6.x, point-in-time recovery support is limited to the following regions:

  `ap-northeast-1`

  `ap-southeast-1`

  `ap-southeast-2`

  `eu-west-1`

  `sa-east-1`

  `us-standard (us-east-1)`

  `us-west-1`

  `us-west-2`

Use the `Previous` button or select a tab to return to a tab to review or update information; when you have completed the `Create a New Server` dialog, click Launch to create the database cluster.

A popup dialog confirms that EDB Ark is creating a new cluster; click the `X` in the upper-right corner of the popup to

close the popup.

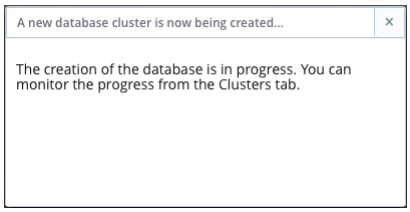


Fig. 5.7: A popup confirms that the new cluster is being created.

Navigate to the `Clusters` tab of the Ark console to monitor the creation of the cluster.

Using a Template to Create a Cluster

If you select the `Launch From Template` option when deploying a cluster or are a *Template only* user, a dialog that offers limited options will open when you deploy a cluster.

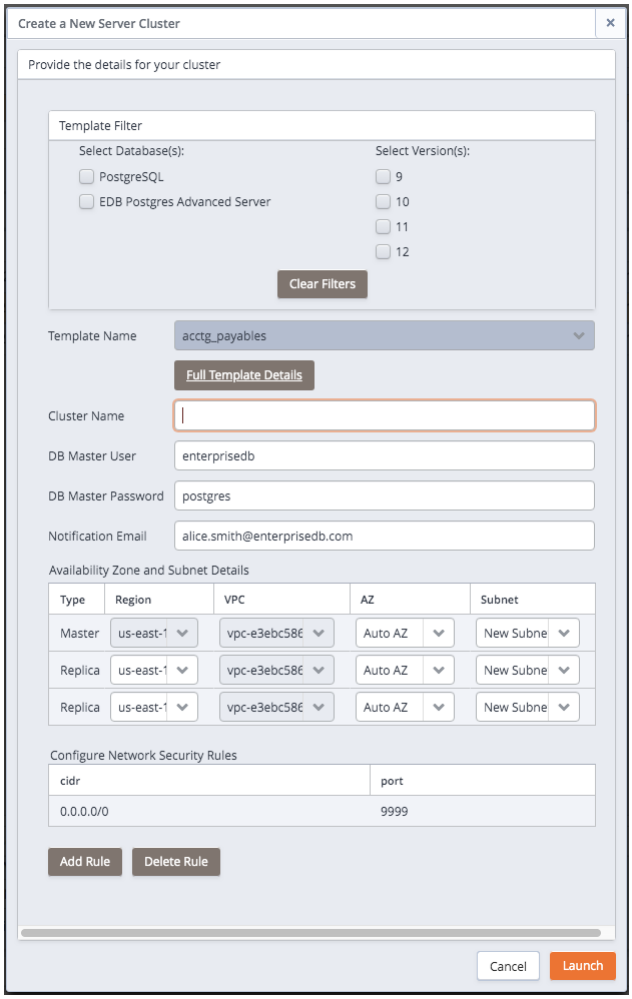


Fig. 5.8: Creating a cluster from a template.

Use fields on the `Create a New Server Cluster` dialog to define your cluster:

- Use the `Template Filter` panel to filter the list of templates that will be made available in the `Template Name` drop-down listbox. Select:
  - The database or databases from which you would like to choose.
  - The version or versions from which you would like to choose.

- Use the `Template Name` drop-down listbox to select the template that you wish to use to configure your cluster. To review template configuration details, select the `Full Template Details` link; the `Template Details` dialog opens.

Fig. 5.9: Reviewing Template details

After selecting the template that you wish to use, use the fields on the dialog to finish defining your cluster:

- Use the `Cluster Name` field to specify a name for the new cluster.
- Use the `DB Master User` field to specify the name of the database superuser.
- Use the `DB Master Password` field to specify the password associated with the database superuser.
- Use the `Notification Email` field to provide the email address that will receive notices about changes to the cluster status.
- Use fields in the `Availability Zone and Subnet Details` section to specify the regions in which the cluster nodes will be deployed:
  - Use the `Region` drop-down listbox to select the region in which the node will be deployed.
  - Use the `VPC` drop-down listbox to select the vpc on which the node will be deployed.
  - Use the `AZ` drop-down listbox to select the availability zone in which the node will reside.
  - Use the `Subnet` drop-down listbox to select the subnet that the node will use.

- If applicable, highlight a rule, and use the `Add Rule` or `Delete Rule` button to define security rules that allow access to the cluster. By default, the load balancer port is open to any IP address for client connections; you may choose to delete the rule, and specify a more restrictive IP range.
  - To delete a rule from the list, highlight the entry and click the `Delete Rule` button; you will be prompted to confirm that you wish to delete the entry before the server removes the rule.
  - Click the `Add Rule` button to open the `Add Rule` dialog and provide access to a port. On the `Add Rule` dialog:
    - Use the `Port` drop-down listbox to select the port that can be accessed from the specified CIDR.
    - Use the `CIDR` field to specify the address (or address range) that will be allowed access to the server through the selected port.

After completing the `Launch From Template` dialog, click the `Launch` button to provision a cluster that conforms to the cluster configuration.

Cloning a Server Cluster

With a few simple steps, you can create a developer sandbox that contains a duplicate of the original master node. To clone a cluster,

navigate to the `Clusters` tab and highlight the name of the cluster you wish to clone; then, click the `Clone` icon from the left margin. If you are not a Template Only user, a dialog will open that allows you to select your deployment method.

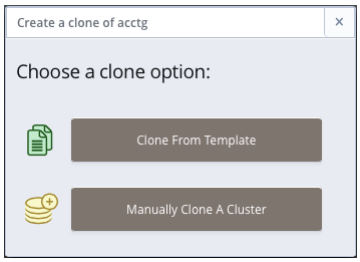


Fig. 5.10: Selecting a Cloning option

If you are a Template Only user or select the `Clone From Template` option on the deployment method selection dialog, a dialog opens that allows you to use a pre-defined template for the cluster configuration; for detailed information about using a template to clone a cluster, see Using a Template to Create a Cluster .

If you select `Manually Clone A Backup`, the dialog shown below opens.

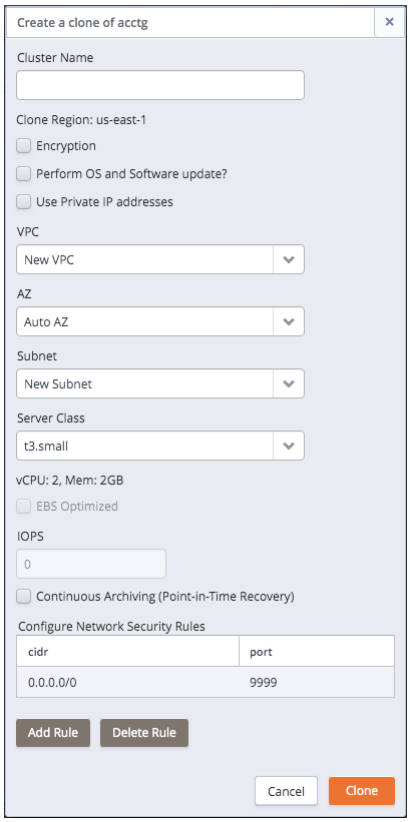


Fig. 5.11: Creating a clone of a database.

When the `Create clone...` dialog opens, provide values in the requested fields:

- Provide a name for the new cluster in the `Cluster Name` field.
- The clone will be created in the `Clone Region` listed below the cluster name.
- Check the box next to `Encryption` if you would like the clone to be created in an encrypted cluster.
- Check the box next to `Perform OS and Software update?` if you would like the server to perform a software update each time the clone is provisioned. Please note: this option is disabled if the database engine is statically provisioned.

- If applicable, check the box next to `Use Private IP addresses` to create the clone your private network.
- If your cluster resides on an Amazon host, use the `VPC` drop-down list box to specify a network name.
- Use the `Availability Zone` drop-down listbox to specify the availability zone in which the new cluster will be created.
- Use the `Subnet` drop-down listbox to select a subnet that will be used by the new cluster.
- Use the `Server Class` drop-down listbox to specify the initial size of the new cluster. The attributes of the selected server class will be listed below the `Server Class` field.
- If your cluster resides on an AWS host, check the box next to `EBS Optimized` to specify that your 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, use the field to specify the level of I/O performance that will be maintained for the cluster by automatic scaling.
- Check the box next to `Continuous Archiving (Point-in-Time Recovery)` to enable point-in-time recovery on the clone.
- Use the buttons in the `Configure Network Security Rules` section to define security rules for the clone.
  - To delete a rule from the list, highlight the entry and click the `Delete Rule` button; you will be prompted to confirm that you wish to delete the entry before the server removes the rule.
  - Click the `Add Rule` button to open the `Add Rule` dialog and provide access to a port. On the `Add Rule` dialog:
    - Use the `Port` drop-down listbox to select the port that can be accessed from the specified CIDR.
    - Use the `CIDR` field to specify the address (or address range) that will be allowed access to the server through the selected port.

When you've completed the dialog, click the `Clone` button to create the sandbox.

When you clone a database, only the master node is recreated in the new cluster; for information about manually adding replica servers to the new cluster, see Manual Scaling.

Using a Template to Clone a Cluster

A clone deployed with a template will be an exact duplicate of the original master node, but will adhere to the cluster deployment rules described in the template by the system administrator. If you are a Template Only user, you will be required to use a template when cloning a cluster. A non-Template Only user may find it easier (especially when cloning a number of clusters) to use a template to define the properties that are common to multiple deployments.

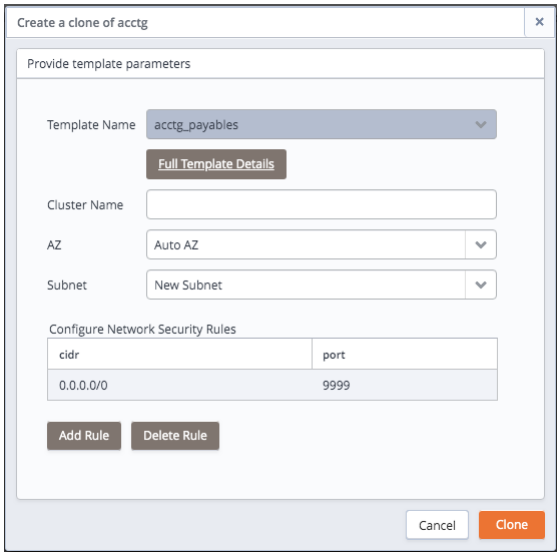


Fig. 5.12: Creating a clone of a database.

When the `Create clone...` dialog opens, provide values in the requested fields:

- Use the `Template Name` drop-down listbox to select a template that will be used for the new cluster; to review the cluster details associated with the template, click the Full Template Details link.
- Provide a name for the clone in the `Cluster Name` field.
- If applicable, use the `Availability Zone` drop-down listbox to specify the availability zone in which the new cluster will be created.
- If applicable, use the `Subnet` drop-down listbox to select a subnet that will be used by the new cluster.

Use the fields in the `Configure Network Security Rules` section to define security rules that allow access to the cluster. By default, the load balancer port is open to any IP address for client connections; you may choose to delete the rule, and specify a more restrictive IP range.

To delete a rule from the list, highlight the rule and click the `Delete Rule` button; you will be prompted to confirm that you wish to delete the entry before the server removes the rule.

Click the `Add Rule` button to open the `Add Rule` dialog and provide access to a port.

Fig. 5.13: Adding a security rule.

On the `Add Rule` dialog:

- Use the `Port` drop-down listbox to select the port that can be accessed from the specified CIDR. A non-administrative user can allow access to ports:

  `9999` - for client connections and load balancing.

  `5432` or `5444` - the cluster specific database listener port.

  An administrative user can use the `Add Rule` dialog to add a rule that allows SSH access to Port `22`.

- Use the `CIDR` field to specify the address (or address range) that will be allowed access to the server through the selected port.

When you're finished, click `Apply` to create the security rule.

After providing the details for the cluster, click the `Clone` button to create the clone of the cluster; select `Cancel` to exit the dialog without creating a cluster.

Modifying a Cluster's Administrative Settings

Fields on the `Administrative Settings` dialog display the current owner and the email address to which notification emails about the state of the cluster are sent. To modify the owner of a cluster or the email address associated with a cluster, highlight the name of a cluster on the `Clusters` tab, and click the `Administrative Settings` icon. The dialog shown below opens.

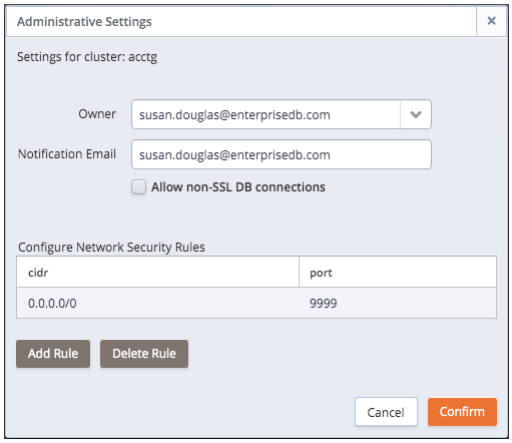


Fig. 5.14: The Administrative Settings dialog

Use the fields on the dialog to modify the administrative settings for the cluster:

- Use the drop-down listbox in the `Owner` field to select a new cluster owner. Please note that only those users with permissions to access the tenant on which the cluster resides are included in the list.
- Use the `Notification Email` field to specify the address to which you wish notices about the state of the cluster to be sent.
- Check the box next to `Allow non-SSL DB connections` to specify that the database can accept connections from a source that is not authenticated via SSL; this will adjust the settings in the `pg_hba.conf` file.

Use the fields in the `Configure Network Security Rules` section to define security rules that allow access to the cluster. By default, the load balancer port is open to any IP address for client connections; you may choose to delete the rule, and specify a more restrictive IP range.

To delete a rule from the list, highlight the rule and click the `Delete Rule` button; you will be prompted to confirm that you wish to delete the entry before the server removes the rule.

Click the `Add Rule` button to open the `Add Rule` dialog and provide access to a port.

Fig. 5.15: Adding a security rule.

On the `Add Rule` dialog:

- Use the `Port` drop-down listbox to select the port that can be accessed from the specified CIDR. A non-administrative user can allow access to ports:

  `9999` - for client connections and load balancing.

  `5432` or `5444` - the cluster specific database listener port.

  An administrative user can use the `Add Rule` dialog to add a rule that allows SSH access to Port `22`.

- Use the CIDR field to specify the address (or address range) that will be allowed access to the server through the selected port.

When you're finished, click `Apply` to create the security rule.

After modifying the configuration details for the cluster, click the `Confirm` button; a dialog will open, prompting you to provide the password associated with the connected session.

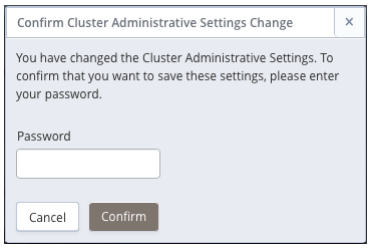


Fig. 5.16: Provide a password to confirm changes.

Provide a password in the `Password` field and click `Confirm` to save your changes and exit, or `Cancel` to exit without saving the changes.

2.6 Connecting to an EDB Ark Cluster

To connect to an Ark cluster, provide the IP address and port of the server, and the credentials associated with the role defined when the server cluster was created.

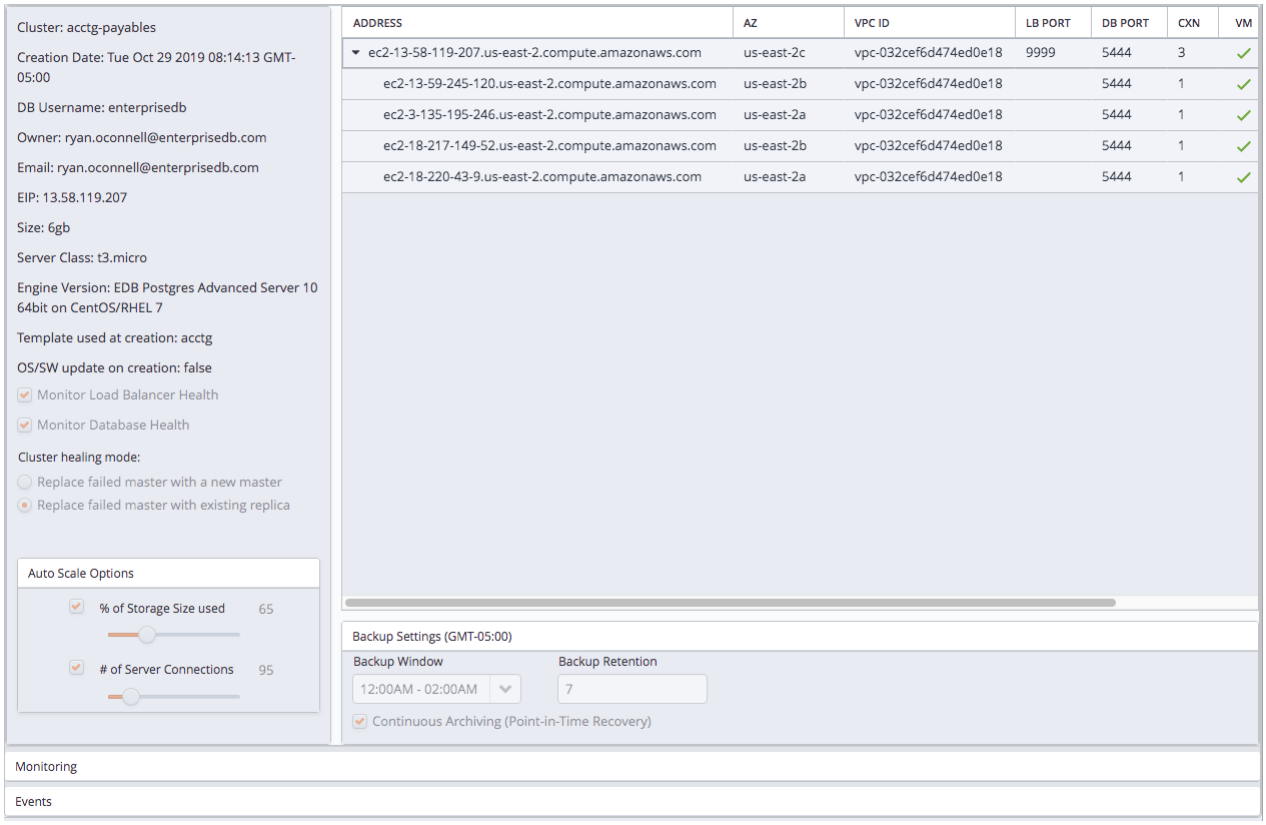


Fig. 6.1: The Details panel on the Clusters tab.

If you have defined a cluster with two or more servers, client applications should always connect to the load balancing port of the master server (the first DNS name listed on the `Details` panel). This will ensure that read requests are

distributed efficiently across the cluster replicas to maximize performance, while write requests are directed only to the cluster master. Replica server nodes are listed below the master node in the tree view.

- The `DNSNAME` column displays the address of the node; a connecting client should use this address when connecting to a specific server.
- The `LB PORT` column displays the port number to which a client application should connect to utilize load balancing.

  Since only the master node of a multi-server cluster operates in read/write mode, all write queries will be directed to the master node, while any read-only queries may be directed to a replica node.
- The `DB PORT` column displays the default listener port for the Advanced Server or PostgreSQL server. To connect directly to the database listener port, modify the cluster's security group to allow connections from your client.

Use the authentication information (`Master User` and `Master Password`) provided on the `Create a New Server Cluster` dialog to establish the initial connection to the cluster as the database superuser. Please note that connecting with this identity grants you superuser privileges on the server; you should not share this connection information with un-trusted users.

After connecting as the database superuser, you should create lesser-privileged user roles with which non-administrative users will connect.

Using ssh to Access a Server

Before you can connect to a cluster node with ssh, an administrator must modify the security group of the cluster node to which you are connecting to permit ssh connections. If the console is federated (and the node resides in a remote region), you must modify the security group on the remote console to allow ssh connections. You can modify the security group via the Administrative Settings dialog on the Ark console or at the AWS console.

EDB Ark creates an ssh key when you create a new cluster; each cluster has a unique key. Before connecting to an Ark instance, 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 as shown below.

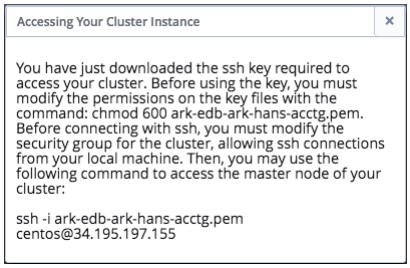


Fig. 6.2: Accessing Your Cluster Instance.

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.

After connecting via ssh, you can:

- Stop, start, or restart the Postgres server.
- Download and install Postgres extensions.
- Use the PostgreSQL Client Applications.
- Invoke PostgreSQL Server Applications.

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>
```

Connecting to EDB Ark with the psql Client

After connecting to a server hosted on EDB Ark with the psql client, you can invoke SQL commands or use meta-commands to:

- Execute queries
- Insert, update, and delete data
- Create and manage database objects (tables, indexes, views, etc.)
- Create user roles and manage privileges
- Review object and role attributes
- Invoke scripts containing complex (or simple) commands

By default, an EDB Ark cluster is only open to connections via port `9999` on the master node. Port `9999` is a good choice if you are connecting for the purpose of querying the database, but if you are modifying database objects, or performing administrative functions, you should connect directly to the server's listener port.

Please note: some administrative functions, if executed over port `9999`, may be directed to the incorrect node of a multi-node cluster where they may not have the intended effect, or may return an invalid value.

The listener port number is displayed in the `DBPORT` column of the `Details` panel of the `Clusters` tab.

Before connecting to the server's listener port, an Ark administrator must modify the security group to allow connections from the host of your client application.

Connecting with psql From a Local Workstation

After installing Advanced Server or PostgreSQL on a local workstation, you can use psql to perform administrative tasks on an EDB Ark cluster.

To open the psql client on an Advanced Server Windows workstation, navigate through the operating system menu to the `EDB Postgres` menu and select `EDB-PSQL`.

To open a psql client on a PostgreSQL workstation, navigate through the operating system menu to the `PostgreSQL` menu, and select `SQL Shell (psql)`.

If you have used a package to install Advanced Server or PostgreSQL, you will find the executable in the `bin` directory under your installation. You can invoke the executable at the command line; for example, to start the client on an Advanced Server host, use the command:

```
/usr/edb/as11/bin/psql -d edb -U enterprisedb
```

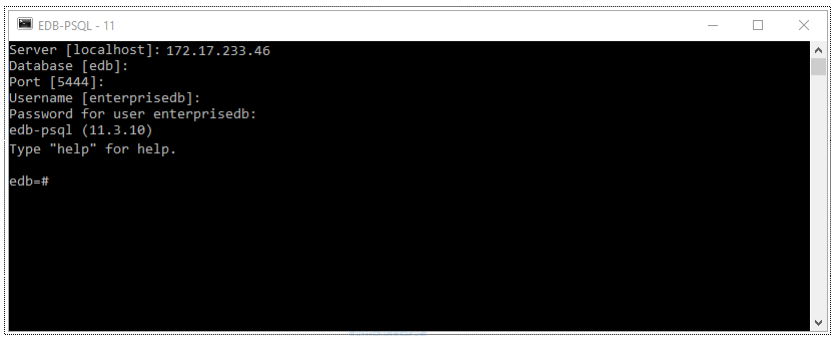


Fig. 6.3: The psql command line utility.

Provide connection information for the server to which you are connecting:

- When prompted for a Server, enter the IP address or DNS name of the EDB Ark server. The IP address is displayed in the `DNSNAME` column on the `Details` panel of the `Clusters` tab of the Ark console.
- When prompted for a Database, enter the name of the database to which you wish to connect. By default, an Advanced Server cluster is created with a database named edb; a PostgreSQL cluster is created with a database named postgres.
- When prompted for a `Port`, enter the port on which the server is listening. For database queries, you can use port `9999`; if you are modifying database objects or performing administrative functions, you should use the server's listener port (5444 for an Advanced Server cluster, 5432 for a PostgreSQL cluster).
- When prompted for a `Username`, enter the role you wish to use when connecting to the server. The name of the database superuser is specified in the Master User field when defining an EDB Ark server cluster. By default, the Advanced Server database superuser is `enterprisedb`. The default superuser of a PostgreSQL database is `postgres`.
- When prompted for a `Password`, enter the password associated with that role. The database superuser's password is specified in the Master Password field when defining an EDB Ark server cluster.

After connecting, the prompt will display the name of the database to which you are connected.

Invoking psql on an EDB Ark Server

To use a copy of the psql client that resides on the EDB Ark host, first connect to the cluster using ssh:

```
ssh -i path <ssh_key> <root@host_name>
```

After connecting to the host, assume the identity of the database superuser (or a user with sufficient privileges to invoke the client). On an Advanced Server host, use the command:

```
sudo su enterprisedb
```

On a PostgreSQL host, use the command:

```
sudo su postgres
```

Then, invoke the psql client. On an Advanced Server host, use the command:

```
/usr/bin/edb-psql -d edb
```

On a PostgreSQL host, use the command:

```
/usr/bin/psql -d postgres
```

Include the `-d` option to specify the name of the database to which you wish to connect. The session opens as shown below.

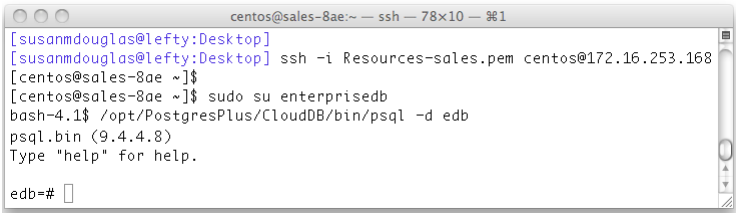


Fig. 6.4: A psql session on the EDB Ark host.

To exit the psql client, enter `\q`.

For information about using psql and the psql meta-commands, please see the Postgres documentation at:

http://www.enterprisedb.com/docs/en/11/pg/app-psql.html

Using iptables Rules

EDB Ark uses iptables rules to manage security on the console and cluster nodes. 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 to the default administration port.
- 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).

2.7 Managing Backups and Recovery

When you use the Ark console to take a backup, EDB Ark makes a copy of the contents of the PostgreSQL `PGDATA` directory. The `PGDATA` directory contains the data and the meta-data required to construct an exact copy of the Postgres data cluster (the data and the database objects that reside within that Postgres instance).

To capture a backup of a cluster, navigate to the `Clusters` tab, highlight a name in the cluster list, and click the `Backup` icon. The `Backup Data?` dialog opens as shown below:

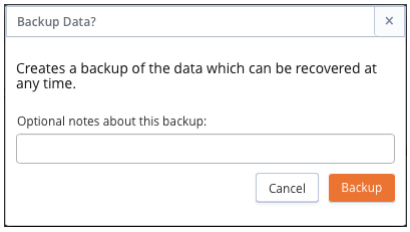


Fig. 7.1: The Backup Data? dialog.

You can include a reference note about the backup that can be viewed on the Backups tab by adding a message to the `Optional notes` field on the `Backup Data?` dialog before clicking the `Backup` button.

When you click the `Backup` button, EDB Ark will perform the backup. While EDB Ark performs the backup, the `PENDING` column of the selected cluster on the `Clusters` tab will display the message: `Backup in progress`.

Performing a Base Backup for Point-In-Time Recovery

When point-in-time recovery is 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. Note that if you deselect this option, the cluster (and subsequent automatic backups) will be re-configured to not include support for point-in-time recovery.

When point-in-time recovery is enabled, the value specified in the `Backup Retention` field of the `Create cluster` dialog determines the duration of the point-in-time recovery backup window. For example, if you specify a value of `7`, the backup window will be 7 calendar days long. When the backup retention threshold is reached, the oldest base backup is removed, as well as any WAL files required to perform a recovery with that backup.

Please note that you cannot perform a base backup on a cluster while the database is in recovery and not accepting connections. If you attempt to perform a base backup during recovery, the backup will fail (the failure will be noted on the `Events` panel of the `Clusters` tab). You should instead wait until the database recovery is complete to enable point-in-time recovery for the cluster.

Point-in-time recovery is enabled on the Details panel of the Clusters tab . If a base backup fails, you can trigger EDB Ark to perform a base backup by disabling point-in-time recovery, and then (after waiting a few minutes) re-enable point-in-time recovery.

Reviewing Stored Backups

Navigate to the Backups tab to review a list of stored cluster backups.

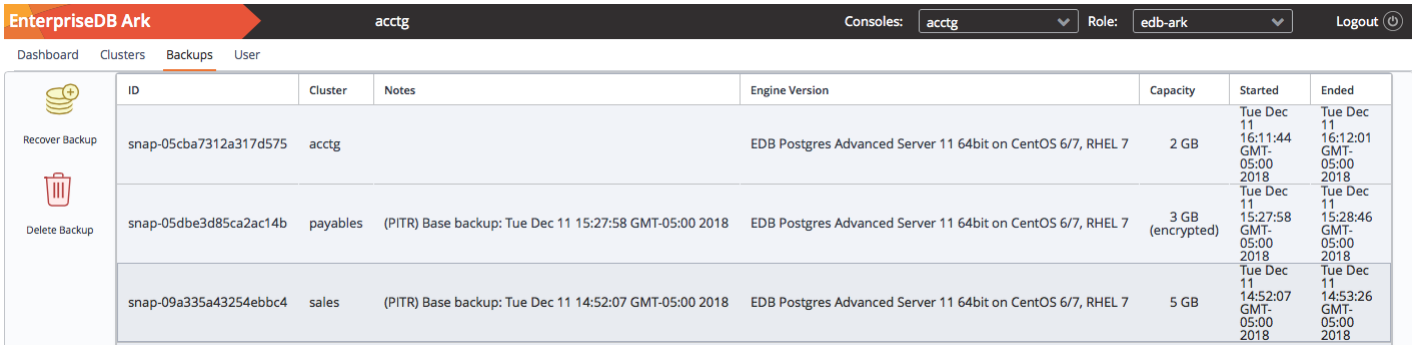


Fig. 7.2: The Backups tab of the Ark console.

A backup captures and stores the status and condition of a cluster at a specific point-in-time.

- The `ID` column contains a unique backup identifier.
- The `CLUSTER` column displays the name of the cluster that was the target of the backup.
- The `NOTES` column displays an informational note (provided by either the user or the system at the time of backup). Those messages that include (PITR) can be restored to a specific point-in-time within the backup window.
- The `ENGINE VERSION` column contains a description of the Postgres version that the saved cluster is using.
- The `CAPACITY` column contains the storage capacity of the cluster at the time that the backup was taken.
- The `STARTED` column displays the date and time that the backup was initiated.
- The `ENDED` column displays the data and time that the backup completed.

You can use the icons on the left side of the Backups tab to restore or delete the selected backup:

Status indicators provide quick visual feedback about each feature:

	Highlight a backup in the list, and click the Recover Backup icon to open a dialog that allows you to restore a cluster from the selected backup.
	Highlight one or more backups in the list and click the Delete Backup icon to delete the selected backups. A popup will ask you to confirm that you wish to delete the selected backups before the backups are actually deleted.

Restoring a Cluster from Backup

You can use a template or manually provide cluster properties when restoring a cluster from a backup. To restore a backup into a new cluster, navigate to the Backups tab and highlight the name of a backup. Then, click the `Recover Backup` icon (located in the left margin).

If you are not a Template Only user, a dialog will open that allows you to select the method by which you will specify cluster preferences.

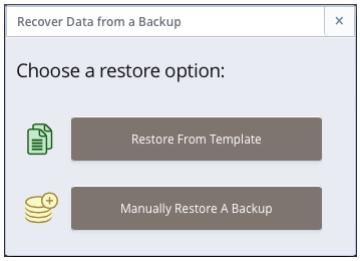


Fig. 7.3: Selecting a Restore option.

If you are a Template Only user or select the `Restore From Template` option on the recovery method dialog, a

dialog opens that allows you to use a pre-defined template for the cluster configuration; for detailed information about using a template to clone a cluster, see Using a Template to Restore from Backup .

If you select `Manually Restore A Backup`, the following dialog opens:

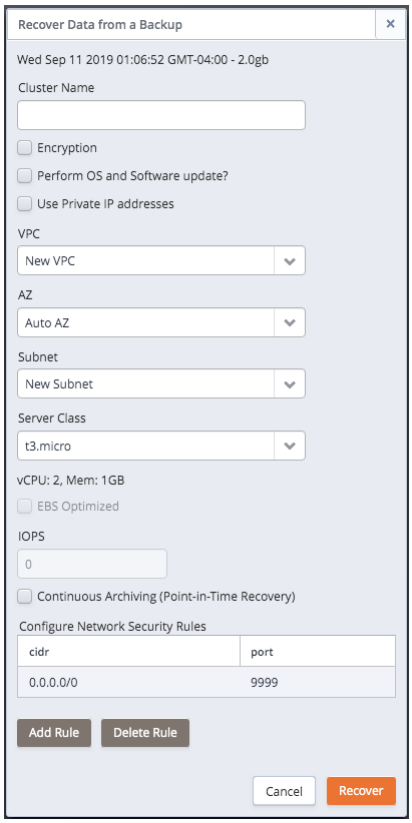


Fig. 7.4: The Recover Data from a Backup dialog.

When the `Recover Data from a Backup` dialog opens:

- If applicable, use the calendar selector in the `Recovery Point` field to specify the recovery target (the date and time that the database was in the state in which you wish the new cluster to start). The Recovery Point field is only displayed for backups that were taken with point-in-time recovery implemented; you cannot perform a point-in-time recovery with a backup unless point-in-time recovery is enabled for the cluster when the backup was taken.
- Specify a name for the new cluster in the `Cluster Name` field.
- Check the box next to `Encryption` to specify that the new cluster should reside in an encrypted cluster. Please note that you can restore a non-encrypted backup into an encrypted cluster.
- Check the box next to `Perform OS and Software` update to instruct EDB Ark to perform a yum update whenever the cluster is provisioned. Please note: this option is disabled if the database engine is statically provisioned.
- If applicable, check the box to the left of `Use Private IP addresses` to restore the backup into a private IP address.
- If your cluster resides on an Amazon host, use the `VPC` drop-down listbox to select a VPC on which the cluster will reside.
- Use the `Availability Zone` drop-down listbox to the right of each node to select the availability zone in which the node will reside.
- Use the `Subnet` drop-down listbox to the right of each node to select the subnet that the node will use.

- Use the `Server Class` drop-down listbox to specify the server class of the new cluster.
- If your cluster resides on an AWS host, check the box next to `EBS Optimized` to specify that your 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, use the field to 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.

  Note that you can increase the IOPS value of your cluster by recovering the cluster from a snapshot into a cluster with a higher value or cloning your database into a cluster with a higher IOPS value.
- Check the box next to `Continuous Archiving (Point-In-Time Recovery)` to indicate that the new cluster should implement point-in-time recovery. Please note that to restore into a cluster with point-in-time recovery enabled, the backup from which you are restoring must have had point-in-time recovery implemented when the backup was taken. The checkbox will not be available if point-in-time recovery was not implemented when the backup was taken.
- Use the `Add Rule` button to open a dialog that allows you to open a port for connections from a specified CIDR formatted address.On the Add Rule dialog:

  Use the `Port` drop-down listbox to select the port that can be accessed from the specified CIDR. A non-administrative user can allow access to ports:

  9999 - for client connections and load balancing.

  5432 or 5444 - the cluster specific database listener port.

  An administrative user can use the `Add Rule` dialog to add a rule that allows SSH access to Port `22`.

  Use the CIDR field to specify the address (or address range) that will be allowed access to the server through the selected port.

  When you're finished, click Apply to create the security rule and continue.
- Highlight a rule and click the `Delete Rule` button to remove a security rule.

Click the `Recover` button to continue, or the `Cancel` button to exit without starting the recovery process. A popup confirms that the cluster is being restored; close the popup and navigate to the `Clusters` tab to monitor the restoration process.

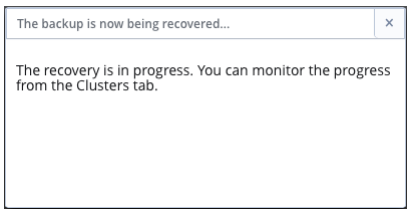


Fig. 7.5: The recovery is in progress.

Please note: when you restore a backup, the server configuration will match the original configuration, but the server addresses will change.

Please note: when restoring a cluster from backup, you may need to modify parameters in the postgresql.conf file on the restored cluster to reflect the available memory of the new instance if the server class has changed from the original setting (the default value in the `Server Class` field). After modifying the server configuration, restart the server for

the changes to take effect.

Using a Template to Restore from Backup

If you are a Template Only user, you will be required to use a template when restoring a backup into a new cluster. A non-Template Only user may find it easier (especially when restoring a number of clusters) to use a template to define the properties that are common to multiple deployments.

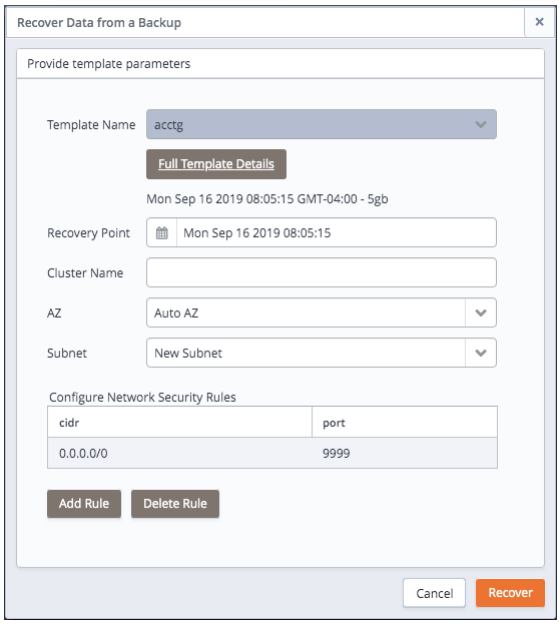


Fig. 7.6: Using a template to restore from a backup.

If you are using a template when restoring a cluster from backup, use the dialog shown to provide the non-template details:

- Use the `Template Name` drop-down listbox to select a template that will be used for the new cluster; to review the cluster details associated with the template, click the Full Template Details link.
- If applicable, use the calendar selector to specify a point-in-time to which to recover.
- Specify a name for the new cluster in the `Cluster Name` field.
- Use the `AZ` drop-down listbox to the right of each node to select the availability zone in which the node will reside.
- Use the `Subnet` drop-down listbox to the right of each node to select the subnet that the node will use.
- Use the `Add Rule` button to open a dialog that allows you to open a port for connections from a specified CIDR formatted address.
- Highlight a rule and click the `Delete Rule` button to remove a security rule.

2.8 Automatic Failover

The EDB Ark cluster manager constantly monitors the state of each cluster. Each cluster is composed of a single master Postgres instance that operates in read-write mode (performing all writes to the database) and one or more replica Postgres instances. Replica nodes are read-only, automatically duplicating all data found on the master node, and all changes made to that data.

If a replica fails, EDB Ark automatically spins up a new replica instance and attaches it to the master database. The

cluster continues operating during the replacement process, with the master servicing writes and reads, and the remaining replicas servicing reads. Overall read performance may degrade for a short period of time until the cluster is returned to full strength.

If a master failover occurs, the server will enforce one of two behaviors, specified by the Cluster healing mode radio buttons, located on the Details panel of the Clusters tab:

- Select the `Replace failed master with a new master` radio button to specify that the cluster manager should create a new master to replace a failed master node.

  When replacing a failed master node with a new master node, the data volumes from the failed instance are attached to the new master node, preserving all transactions that were committed on the master.

- Select the `Replace failed master with existing replica` radio button to specify that the cluster manager should promote a replica node to be the new master node for the cluster. Choose this option when speed of recovery is important, and your application can tolerate the loss of some transactions. This is the default behavior.

  When replacing a failed master node with an existing replica, a replica node is marked for promotion to master node, while the other replica nodes are re-configured to replicate data from the new master node. Since replica nodes use asynchronous replication, any data that was committed to the old master node, but not yet pushed to the replica prior to the node failure will be lost.

  If you opt to promote a replica to replace the master node, a replacement replica will also be added to the cluster during the failover process, returning the cluster to full strength. This self-healing property is at the heart of providing high availability to cluster users.

Please note that replacing a failed master node with a new master node can take a bit longer than promoting a replica node to the role of master, but it does have the advantage of guaranteeing that no committed data will be lost.

Triggering a Failover

By design, EDB Ark *does not* perform a failover when the Postgres server is stopped, because the server stop or restart may be intentional:

- A user may intentionally restart the server when performing maintenance or tuning. For example, a server restart is required when updating server configuration parameters; this restart will not invoke failover.
- If a user intentionally kills the postmaster process, the server will not failover; the postmaster process is responsible for restarting the server.
- The Postgres server may intentionally perform a server restart. For example, when a backend server process crashes (or is intentionally killed by a user), the Postgres server automatically invokes a restart.

When a failover is complete, EDB Ark does not delete the original master instance of the database; you can use the preserved master instance to perform any post-mortem activities that may be required. If you do not wish to utilize the preserved instance, you should use the management console to delete the instance.

Manually Promoting a Standby Node

You can use the `Promote` option on a cluster's context menu to manually promote a standby node to master node. When you promote the standby, the master node is deleted. You must initiate the promotion from a console that resides in the same region as the standby that is being promoted.

To promote a node to master, right-click on the cluster name in the `Details` panel, and select `Promote` from the context menu.

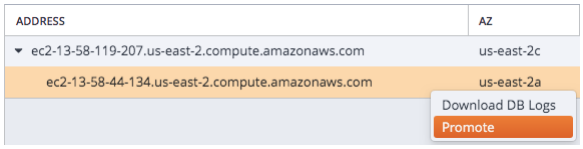


Fig. 8.1: The Promote option on a context menu

When you select the `Promote` option, the console will ask you to confirm that you wish to promote the standby to master and delete the original master node.

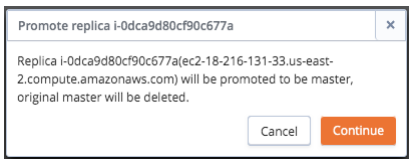


Fig. 8.2: Confirming that you wish to promote a standby

Click `Continue` to instruct Ark to perform the promotion. During the promotion, status messages will inform you of the state of the cluster.

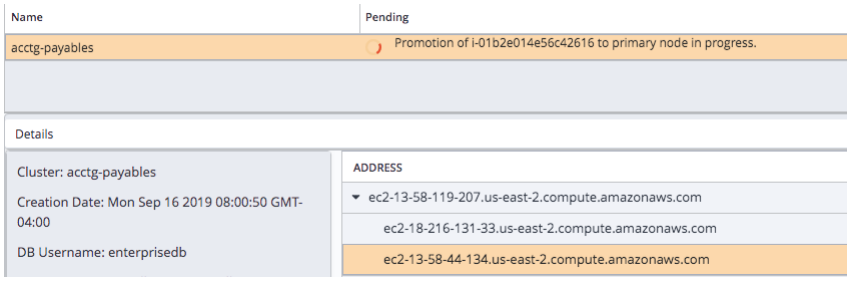


Fig. 8.3: Status messages display the state of the cluster

2.9 Manual Scaling

The Ark console makes it simple to add replicas and storage to an existing cluster, or to upgrade to a larger server class (i.e. vertical scaling).

- Adding additional replicas to your database cluster increases the CPU power available to handle additional client requests or applications, increasing the number of client connections that can be serviced. When the scale up is complete, each additional replica automatically assumes a share of the read-only workload from incoming queries.
- Adding additional storage to the cluster increases the amount of data that can be stored by the database servers. When you add additional storage to the cluster, each member of the cluster gets the additional storage amount.
- Vertically scaling to a larger server class increases the processing capabilities of your cluster, allowing the server to process customer requests with greater speed.

You can also downsize a cluster by selectively removing a replica. You can machine scale to a smaller machine type to reduce resource usage (cpu/memory) and/or cost.

Please note: if you are a Template Only user, access to automatic scaling behaviors is determined by the configuration specified on the template used to deploy your cluster.

Manually Adding Replicas and Storage

EDB Ark's `Scale Up` dialog makes it simple to manually add additional replicas to a cluster if you find that server resources are strained. The dialog also allows you to increase the amount of storage available to a cluster.

If you specify that EDB Ark should add both storage and replicas, EDB Ark will process the request for additional storage *before* adding replicas to the cluster. All of the nodes on the cluster will be of the newly specified storage size.

To add a replica or storage space to a cluster, navigate to the Clusters tab , highlight a cluster name, and select the `Scale Up` icon. The `Scale Up` dialog opens as shown below:

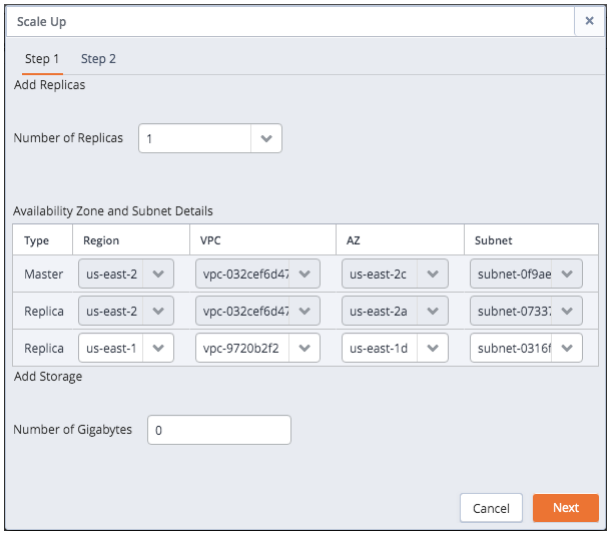


Fig. 9.1: The Scale Up dialog.

Use the drop-down listboxes on the `Step 1` tab to specify:

- The number of replicas to add to the cluster.
- The region in which each node of the cluster will be provisioned.
- The VPC in which the node will be provisioned.
- The AZ (availability zone) in which the node will be provisioned.
- The subnet that will be used by each node of the cluster.
- The amount of storage memory (in Gigabytes) that will be added to each server in the cluster.

When you've completed the dialog, click `Next` to continue to the `Step 2` tab:

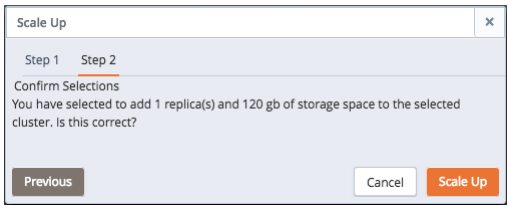


Fig. 9.2: The Step 2 tab.

Use the `Previous` button to return to the `Step 1` tab to modify specified values. Use the `Scale Up` button to confirm that you wish to add the specified number of replication servers or the specified amount of memory to the cluster. Use the `Cancel button`, or simply close the dialog to exit without modifying the cluster.

When scaling begins, EDB Ark will confirm that replicas or memory are being added to the cluster.

Manually Removing a Replica

EDB Ark's `Scale Down` dialog makes it simple to manually remove one or more unneeded replicas from a cluster.

To delete a replica, navigate to the Clusters tab , and click the `Scale Down` icon. The `Scale Down` dialog opens as shown below:

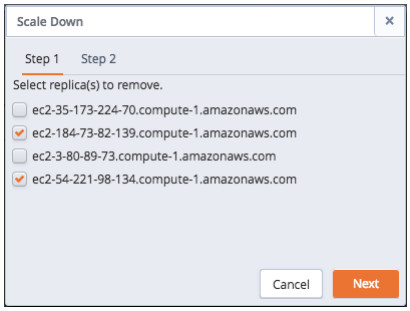


Fig. 9.3: The Scale Down dialog.

Check the box to the left of the name of a replica, and click `Next` to proceed to the `Step 2` tab of the dialog.

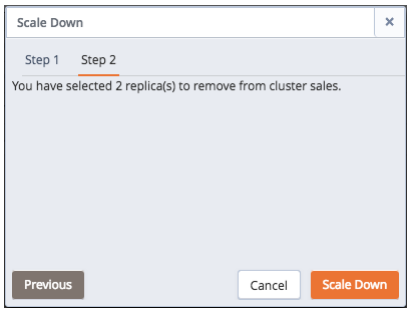


Fig. 9.4: The Step 2 tab

Click `Scale Down` to confirm that you wish to remove the replica, or Previous to return to the `Step 1` tab. Use the `Cancel` button, or simply close the dialog to exit without modifying the cluster.

Manually Changing the Server Class

When your RAM processing needs, CPU power, or other circumstances warrant a larger virtual machine for your application, you can vertically scale to a larger server class. You can either:

- Use the `Scale Machine Type` dialog to copy the cluster into a larger server class.

  When you use the `Scale Machine Type` dialog to move your cluster into a larger server class, you must provide a new name for the upgraded cluster. You can also use the dialog to specify that EDB Ark should re-assign the IP address of the cluster, so the upgrade will be transparent to connecting clients.

  Please note: you may wish to postpone the IP address reassignment to perform configuration tasks or test the new server size.

- Use the `pg_dump` and `pg_restore` utilities to move the cluster into a larger server class.

  To move to a larger server class, use the pg_dump utility to make a backup of the cluster. After backing up the cluster, create a new instance with the larger server class, and use pg_restore to restore the cluster on the new instance. For information about using pg_dump and pg_restore, see Moving an Existing Database into a New Cluster

.

You can also downsize a cluster by selectively removing a replica. You can machine scale to a smaller machine type to reduce resource usage (cpu/memory) and/or cost.

When you vertically scale your cluster with the `Scale Machine Type` dialog, EDB Ark will copy the existing cluster into a new cluster of a different server class, and optionally re-assign the IP address of the existing cluster to the new cluster. To open the `Scale Machine Type` dialog, navigate to the Clusters tab and select the `Scale Machine Type` icon.

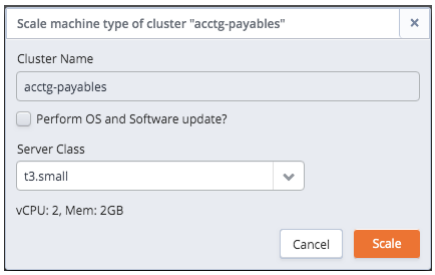


Fig. 9.5: The Scale Machine Type dialog.

Use the fields on the `Scale Machine Type` dialog to specify details about the new cluster:

- Check the box next to `Perform OS and Software update` to instruct EDB Ark to perform a yum update whenever the cluster is provisioned. Please note: this option is disabled if the database engine used to provision the cluster is statically provisioned.
- Use the `Server Class` drop-down listbox to specify the size of the new cluster.

  Please note that if you are a Template Only user, the types listed in the `Server Class` drop-down listbox will be limited to those types that are included in template definitions for the current tenant.

  When you click the `Scale` button to start scaling the cluster, EDB Ark will confirm that the scaling is in progress.

  Before creating the new cluster, EDB Ark will perform a backup of the original cluster. During the process, status indicators in the `PENDING` column of the `Clusters` tab will keep you informed as EDB Ark backs up the original cluster, and initializes the new cluster.

2.10 Automatic Scaling

Adding additional replicas to your database cluster increases the number of client connections and queries that each cluster can handle, while maintaining a high-level of overall performance. Each additional replica automatically assumes a share of the read-only workload from incoming queries.

When auto-scaling in enabled, EDB Ark monitors the server storage and connection resources in use, and automatically adds additional resources when usage exceeds a user specified percent.

- When the `% of Storage Size used` is reached, EDB Ark will automatically increase your data space by 50%.
- When the `# of Server Connections` is reached, EDB Ark adds replica nodes.

Please note: if you are a Template Only user, access to automatic scaling behaviors is determined by the configuration specified on the template used to deploy your cluster.

Adjusting the Automatic Scaling Thresholds

Use the `Auto Scale Options` controls to adjust the threshold at which EDB Ark automatically scales up cluster resources. The `Auto Scale Options` controls are located on the `Details` panel; to access the `Details` panel, navigate to the Clusters tab, and highlight the name of a cluster.

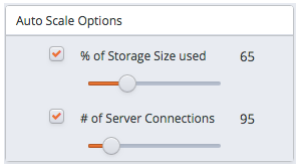


Fig. 10.1: The Auto Scale Options controls.

Adjust the `Auto Scale Options` sliders to increase or decrease the thresholds at which automatic scaling is invoked. When you modify the values, EDB Ark will display a `New Value Saved` notice, alerting you that your changes have been saved.

2.11 Load Balancing

EDB Ark uses pgPool functionality to implement automatic load balancing. Load balancing increases system performance by distributing client queries to replica nodes, while routing database modifications to the master node. Any modifications to the master node are subsequently propagated to each replica using Postgres streaming replication.

Utilizing Load Balancing

By default, load balancing is enabled on an EDB Ark cluster. To utilize load balancing, you should direct client applications to connect to the load balancing port (by default, 9999). A cluster's load balancing port number is displayed in the LBPORT column on the Details pane of the Clusters tab of the Ark console.

pgPool may direct the following statement types to *either* a primary or a standby node:

- SELECT statements (not listed below)
- COPY TO
- DECLARE
- FETCH
- CLOSE
- SHOW
- SET
- DISCARD
- DEALLOCATE ALL

When deciding which node a query should be routed to, pgPool checks the transaction log number; if the transaction log number on the standby server is lower than the log number on the master, pgPool routes the statement to the master node. This helps to ensure that the data returned by the query is the most recent available.

In some cases, specific clauses within a query statement will signal pgPool to direct a statement to the master node. In other cases, the transaction type, or order of commands within a transaction can direct a statement to the master node. By default, the following transaction types will always be executed on the master node:

- SELECT INTO, SELECT FOR UPDATE or SELECT FOR SHARE statements

- SELECT statements within SERIALIZABLE transactions
- SELECT statements that follow an INSERT statement
- SET SESSION CHARACTERISTICS AS TRANSACTION... READ WRITE statements
- SET transaction_read_only = off statements
- EXPLAIN and EXPLAIN ANALYZE SELECT statements
- START TRANSACTION... READ WRITE statements
- LOCK commands that are stricter than ROW EXCLUSIVE MODE
- Transactions that start with a BEGIN statement
- The nextval() and setval() sequence functions
- Large objects creation commands

Please Note: If your application uses JDBC, and the autocommit option is set to false, the JDBC driver will include a BEGIN and COMMIT statement with each SELECT statement. To enable load balancing when using the JDBC driver, your application must include a call to setAutoCommit(true).

pgPool directs the following non-query statement types to the master node only:

- INSERT
- UPDATE
- DELETE
- COPY FROM
- TRUNCATE
- CREATE
- DROP
- ALTER
- COMMENT
- PREPARE TRANSACTION
- COMMIT PREPARED
- ROLLBACK PREPARED
- LISTEN
- UNLISTEN
- NOTIFY
- VACUUM

Selectively Enforcing Load Balancing

pgPool does not enforce load balancing for `SELECT` statements with a leading white space or leading comment. For example, the following statement would be directed to the master node:

```
/*Ignore load balancing*/ SELECT * FROM emp;
```

To enforce load balancing of `SELECT` statements with leading white space or comments, modify the `pgpool.conf` file, and set the `ignore_leading_white_space` parameter to `true`.

You can also use the `black_list` and `white_list` parameters (located in the `pgpool.conf` file) to instruct pgPool to direct specific statements or functions to the master node. This is useful for cases where a `SELECT` statement (normally directed to a replica) calls a function that in turn might modify the database, and so should be directed to the master.

Monitoring Load Balancer Health

By default, EDB Ark monitors the health of the load balancer to ensure that service is not interrupted. If the load balancer (pgpool) should fail while monitoring is enabled, PgPool will be automatically restarted. If the load balancer cannot be automatically restarted, EDB Ark will display a warning sign next to the cluster name on the Details panel and send a notification email to the cluster user.

Deselect the Monitor Load Balancer Health checkbox (located on the Details panel of the Clusters tab) to indicate that

you do not wish for load balancer health to be monitored and automatically restarted if an interruption in service is detected.

2.12 Customizing Your Cluster

EDB Ark creates fully-functioning, high-availability database clusters of various sizes complete with replication, load balancing, connection pooling, backup and failover capabilities. An EDB Ark cluster can be defined in minutes without any special database knowledge or skills. This characteristic is greatly appreciated by application developers who want to create robust, data-intensive applications quickly, and who may not have the time, inclination, or skills to otherwise achieve the same results. This type of black box setup was designed to dramatically increase the productivity of developers, DBAs, and system administrators alike.

However, there are many users who, while enjoying the black box benefits described above, prefer to take a more hands-on approach to managing their databases. EDB Ark was also designed with these users in mind.

You can also use supporting components to extend the functionality of your EDB Ark cluster; the following sections provide an overview of how to add an extension to a new or existing cluster.

!!! Note The EDB Ark Administrator's console provides an easy way to install and maintain the latest server-related packages. Talk to your system administrator about automatically including supporting components for your cluster when provisioning the database engine.

Adding an Extension to a New Cluster

Supporting components and utilities can extend the functionality of your Postgres cluster. For example, you may want to consider adding EDB Postgres Enterprise Manager for management, monitoring, and statistical analysis functionality, or PostGIS, to provide support for spatial data types and functions.

An administrative user can use the `Optional Node Packages` field on the `Add Engine` or `Edit Engine` dialog to modify a database engine definition, providing the names of optional rpm packages that will be installed (from the specified repository) during provisioning. All engines created with that definition will contain the new component; the component will be provisioned on each replica as well as on the master node. As each rpm is installed, yum will satisfy the dependencies for the new component.

Packages added via the `Optional Node Packages` field on the master node of the cluster will 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 also require manual configuration.

For information about modifying a database engine to add a supporting component, see the *EDB Ark Administrative User's Guide*.

2.13 Database Management

This section details some of the tasks that are performed outside of the Ark console's graphical interface:

- Moving an existing database into an EDB Ark cluster
- Manually modifying configuration parameters on a cluster
- Stopping and starting the server

Moving an Existing Database into a New Cluster

You can use the Postgres `pg_dump` utility to migrate an existing Postgres database (schema, data, and associated database objects) into an EDB Ark cluster.

`pg_dump` creates an archive that contains the commands needed to re-create and populate your existing database. After moving the archive to the master node of the Ark cluster, use pg_restore to uncompress and play the SQL commands contained in the archive. The following section will walk you through the process of moving a database to EDB Ark using pg_dump.

You can also use the `pg_dumpall` utility to move an entire *Postgres* cluster (data, schema information, and roles) to EDB Ark; for detailed information about using pg_dumpall, please see the Postgres documentation at:

http://www.postgresql.org/docs/11/static/app-pg-dumpall.html

The following example assumes that you have provisioned an EDB Ark cluster and opened a port for SSH connections.

Step One – Navigate into the directory that contains pg_dump

Open a terminal window on the system that contains your Postgres source database, assume the identity of the Postgres superuser, and navigate into the bin directory that resides under your Postgres installation directory.

On Advanced Server the path to the `bin` directory is:

```
/usr/ppas-9.x/bin
```

On PostgreSQL, the path to the directory is:

```
/usr/pgsql-9.x/bin
```

Step Two - Create the pg_dump Archive

Use the `pg_dump` utility to create an archive that contains the commands required to recreate a database. When invoking pg_dump, include the -Ft flag to instruct pg_dump to format the output as a tar file, and the -U flag to specify the name of the database superuser:

```
pg_dump -Ft -U <db_superuser> <db_name> > <archive_name.tar>
```

Where:

`db_superuser` is the name of a Postgres database superuser.

`db_name` is the name of the database that you wish to move to EDB Ark.

`archive_name.tar` is the complete path and name of the archive. Please note that you must have permission to write a file to the location specified.

If prompted, enter the password associated with the database superuser.

Step Three - Move the Archive to EDB Ark

Use the scp command to copy the archive to the master server in the EDB Ark cluster; include the `-i` option to specify the location of your ssh key:

```
scp -i <ssh_key_file> <file_name> <user_name@host_name:target>
```

Where:

`ssh_key_file` specifies the pathname of the EDB Ark ssh private key file.

`file_name` specifies the archive name.

`user_name` specifies the name used to connect to the master node of the EDB Ark cluster.

`host_name` specifies the host name of the master node of the EDB Ark cluster; the host name is located on the Details panel of the Clusters tab in the Ark console.

`target` specifies the name of the target directory on the EDB Ark host. Including `:/tmp/` at the end of this command directs `scp` to copy the file to the `tmp` directory

Step Four - Connect to EDB Ark with ssh

Use ssh to connect to your EDB Ark cluster master node. Provide the user identity of the operating system superuser, and the location of the ssh key (on your local host) in the command:

```
ssh -i/<path>/<ssh_key.pem root@host_name>
```

Where:

`path` specifies the location of your EDB Ark ssh certificate on the system from which you are connecting.

`ssh_key.pem` specifies the name of the EDB Ark ssh private key file.

`host_name` specifies the host name of the master node of the EDB Ark cluster; the host name is located on the Details panel of the Clusters tab in the Ark console.

Step Five – Navigate into the bin directory on the Ark Host

After connecting, assume the identity of the database superuser and navigate into the directory on the Ark host that contains the pg_restore utility. On an Advanced Server host:

```
cd /opt/PostgresPlus/CloudDB/bin
```

On a PostgreSQL host:

```
cd /opt/PostgreSQL/CloudDB/bin
```

Step Six - Invoke pg_restore on the master server in the EDB Ark cluster

Before invoking the `pg_restore` utility, you must create the target database in the master server; you can use the createdb client utility at the command line to create the target:

```
createdb -U <db_superuser> <database_name>
```

Where:

`db_superuser` specifies the name of the database superuser. On an Advanced Server cluster, the default is `enterprisedb`; on a PostgreSQL cluster, the default is `postgres`.

`database_name` specifies the name of the database on EDB Ark.

Then, invoke the `pg_restore` utility:

```
pg_restore -Ft -U <db_superuser> /<path>/<archive_name.tar> -d <target_db_name>
```

Where:

`db_superuser specifies` the name of the database superuser. On an Advanced Server cluster, the default is enterprisedb; on a PostgreSQL cluster, the default is `postgres`.

`path` is the pathname to the archive on the Ark.

`archive_name.tar` is the name of the archived database.

`target_db_name` is the name of the target database on the Ark.

Include:

the `-Ft` flag to specify that the file is an archive

the `-U` flag to specify the name of a database superuser.

the `-d target_db_name` flag to specify the name of the target database

Step Seven - Confirm that the Move was Successful

After performing the restore, you can use the psql client to connect to the EDB Ark and confirm that the database has been transferred:

```
psql -U <database_superuser> -d <target_db_name>
```

Where:

`db_superuser specifies` the name of the database superuser. On an Advanced Server cluster, the default is `enterprisedb`; on a PostgreSQL cluster, the default is `postgres`.

`target_db_name` is the name of the target database.

Use the `\dt` command to view a list of database objects in the current database.

To exit the psql client, enter `\q`; to exit the ssh session, type `exit` and `Return`.

For more information about using the psql client, please see the PostgreSQL core documentation .

For more information about using PostgreSQL utilities to move an existing database into EDB Ark, please see the PostgreSQL core documentation .

Manually Modifying Configuration Files

Many of the features of a Postgres server may be influenced by settings specified in configuration files:

- The `postgresql.conf` file determines Postgres server behavior as it pertains to auditing, authentication, file locations, resource usage, query planning, statistic gathering, error handling and more.
- The `pg_hba.conf` file controls the type of authentication that should be used when a client application connects to an EDB Ark service. By default, the pg_hba.conf file is configured to require clients to provide a valid md5-

encrypted password.
- The `pg_ident.conf` file contains user mappings for external authentication methods (like LDAP or GSSAPI). Each entry within the pg_ident.conf file maps an external user name to his corresponding Postgres user name.
- The `pgpool.conf` file determines the behavior of EDB Ark as it pertains to load balancing.

To modify a configuration file:

1. ssh to the node of the cluster that contains the file you wish to modify.
2. Use your choice of editor to modify the files.
3. Reload or restart the server.

When you add or remove nodes from a cluster, EDB Ark takes a backup of your `pg_hba.conf` and `pgpool.conf` configuration files. Configuration file backups are appended with the date that the backup was taken and a unique identifier; for example, `pg_hba.conf.20140319-140903` identifies a backup of the `pg_hba.conf` file.

When modifying a configuration file, you should make changes only to those files that *are not* appended with a timestamp and identifier.

Best Practices for Modifying Configuration Files

Please note that changing parameter settings can have unintended consequences, ranging from degraded performance to system crashes. Consequently, we recommend that only an advanced user who accepts these risks, and has experience with both Postgres and cloud environments modify parameter settings.

There are several ways that you can minimize the risks involved when making parameter changes:

- Always make a snapshot backup of your data before making parameter changes.
- Always use a test cluster to test parameter changes, to ensure they have the intended effect before deploying them to your production environment. Create a test environment that mirrors the final target environment as much as possible - this is easy to accomplish by restoring a production backup into a similar size cluster as the original.
- Only change one parameter at a time (or as few as possible when dealing with interdependent settings) and monitor its effect until you are comfortable with the result.
- Make parameter changes on a *copy* of the existing configuration that is in use for the master or replicas. That way, if the parameter change proves detrimental, it will be easy for you to re-apply the original settings. Keep a backup of the original configuration, so they can be easily restored if necessary.

When adjusting parameters, be mindful of that fact that the master node in the cluster processes both read and write requests, while the replica nodes in the cluster accept only read requests. You can tune the master node and the replica nodes independently to quickly have an impact (either positive or negative) on your write or read performance.

For more information about modifying Postgres server parameters, please see the PostgreSQL core documentation .

Controlling the Database Server

You can use your platform-specific service controller to control a Postgres database. The service name of the database server in an Ark cluster is clouddb.

Controlling a Service on CentOS 7.x

If your cluster resides on CentOS version 7.x, you can use the systemctl command to control the service. The systemctl command must be in your search path and must be invoked with superuser privileges. To use the command, open a command line, and enter:

```
systemctl <action> clouddb
```

Where `action` specifies the action taken by the service command. Specify:

- `status` to discover the current status of the service.
- `start` to start the service.
- `stop` to stop the service.
- `restart` to stop and then start the service.

Controlling a Service on CentOS 6.x

On CentOS version 6.x, you can control a service at the command line with the service command. The Linux service controller mechanism allows you to start and stop the server gracefully. The command must be in your search path and must be invoked with superuser privileges. Open a command line, and issue the command:

```
service clouddb <action>
```

Where `action` specifies the action taken by the service command. Specify:

- `status` to discover the current status of the service.
- `start` to start the service.
- `stop` to stop the service.
- `condstop` to stop the service without displaying a notice if the server is already stopped.
- `restart` to stop and then start the service.
- `condrestart` to restart the service without displaying a notice if the server is already stopped.
- `try-restart` to restart the service without displaying a notice if the server is already stopped.

Updating the Server Version on the EDB Ark Cluster

When an update becomes available for a package installed on your cluster, the Ark console will display an alert symbol in the `UP` column of the `Details` panel for the cluster, and in the `UP` column of the DNSNAME table adjacent to the node that requires an update:

ADDRESS	AZ	VPC ID	LB PORT	DB PORT	CXN	VM	HA	DB	UP
ec2-13-58-119-207.us-east-2.compute.amazonaws.com	us-east-2c	vpc-032cef6d474	9999	5444	2	✓	✓	✓	⚠
ec2-13-58-44-134.us-east-2.compute.amazonaws.com	us-east-2a	vpc-032cef6d474		5444	1	✓	✓	✓	⚠

Fig. 13.1: The DNSNAME table.

The overall cluster status (displayed in the top section of the `Clusters` tab) is based on the values of the nodes within the cluster.

- If all of the nodes within the cluster are up-to-date, the `UP` column displays a green checkmark.
- If one or more nodes require a non-critical update, the `UP` column displays a yellow alert symbol.
- If one or more nodes require a critical update, the `UP` column for the cluster displays a red error symbol.
- If one or more nodes have an unknown package status, the `UP` column for the cluster displays a grey checkmark.

You can use the `Upgrade` icon (located on the Clusters tab) to access a dialog that allows you to update the server version on each node within the cluster.

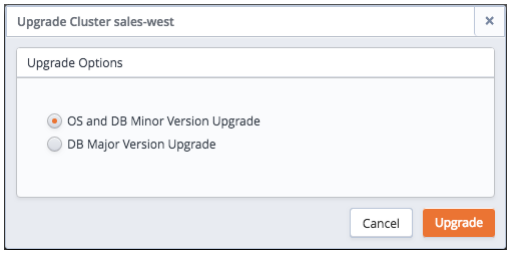


Fig. 13.2: The cluster upgrade dialog.

Select the radio button next to an option to:

- Select `OS and DB Minor Version Upgrade` to invoke a `yum update` command and update any outdated packages and perform a minor database version upgrade on each node of the cluster.
- Select `DB Major Version Upgrade` to select a version and perform a major version upgrade of the server. Please note that this functionality is restricted to users that are not required to use a template when deploying a cluster.

After making a selection, click `Upgrade` to continue.

Performing a Minor Version Upgrade

If you select the radio button next to `OS and DB Minor Version Upgrade` and click the `Upgrade` button, the Ark console will invoke the yum update command on each node of the cluster. The `yum update` command will update all installed packages to the most recent version available of the same release (i.e., if you are running a 9.6 database server, yum will update your database server to the most recent version of 9.6).

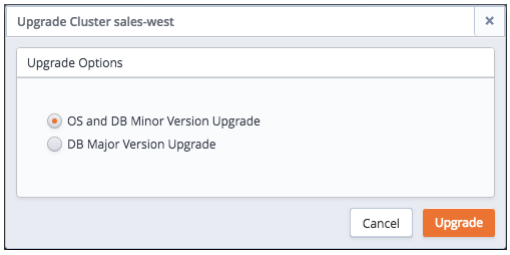


Fig. 13.3: The Upgrade Cluster dialog

Before performing the update, EDB Ark will perform a backup. During the upgrade process, all clients will be disconnected from the server. The updated server will retain the IP address used by the original server. When the update has completed, clients may once again connect.

After performing a yum update, the node will be rebooted, initiating any kernel updates required. When the update completes, EDB Ark will send an email notification that contains a list of the updated packages.

If one or more nodes in your cluster are currently displaying an unknown status, EDB Ark will display an error message. You must correct the problem that is causing the unknown status before EDB Ark can perform an update.

Please note that if the yum update command fails during the upgrade process, EDB Ark will terminate the process and yum update will not be run on any remaining nodes, leaving the cluster partially upgraded.

Performing a Major Version Upgrade

You can use the `Upgrade Cluster` dialog to upgrade the Postgres server installed on your Ark cluster; the upgrade

must be to a more recent version of the same server type and must use the same server image as the current database engine. For example, you may upgrade an Advanced Server version 9.6 database server that resides on a CentOS 6.x host to Advanced Server 11, but you cannot move the server onto a CentOS 7.x host. Similarly, you may not upgrade a PostgreSQL 9.6 database server that resides on a CentOS 7.x host to use an Advanced Server 11 server on a CentOS 7.x host. The server type and host operating system version must remain the same. Please note:

- a major version upgrade may not be performed by a template-only user.
- if over half of the data space allocated to a cluster is used, you must add storage to the cluster before performing the upgrade.

To upgrade a running cluster, select the radio button next to `Major Version Upgrade`, and then use the drop-down listbox to select a server version. Click `Upgrade` to continue.

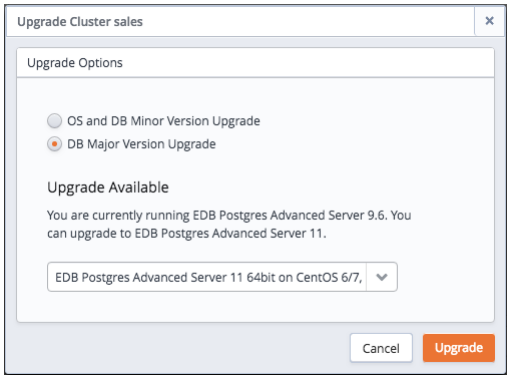


Fig. 13.4: The Upgrade Cluster dialog.

A popup will open, asking you to confirm that you wish to upgrade the server; click the `Upgrade` button to perform an upgrade. The server will be briefly unavailable during the upgrade process. The upgrade does not change the IP address and listening port of the server.

2.14 Troubleshooting

This section provides helpful troubleshooting information; if you still have unanswered questions after reviewing this section, you can also find solutions through EnterpriseDB:

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

in the Customer Portal: http://www.enterprisedb.com/support

via email: 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, you can view your support options at:

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

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

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

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

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

Frequently Asked Questions

Problem: Logging into the Console sometimes takes a long time.

This can be attributed to delays in the connection time to the cloud provider. When you log in, the Console Manager must pass your credentials to the provider to log in; any delays at the service provider may slow your connection time.

Problem: I am attempting to connect to my cluster, but don't know my default database name.

- The name of the default database in an Advanced Server cluster is `edb`.
- The name of the default database in a PostgreSQL cluster is `postgres`.

Problem: unable to connect to the load balancing port (9999).

If you are having difficulty connecting to the load balancing port, you should:

- Make sure you are connecting to the master server's DNS name, rather than a replica's DNS name; the load balancer resides on the master node of an EDB Ark cluster.
- Make sure that your client application is providing an MD-5 encrypted password when attempting to connect to the load balancing port. The username:password-md5 combination is stored in `pgpool_passwd.conf`, and is automatically updated when a user changes password, or when a new user is created.

Problem: pgpool keeps issuing the following error: make_persistent_db_connection: s_do_auth failed.

pgpool attempts to connect to each node to perform replication lag checking. This happens unconditionally if pgpool is configured in a master-slave mode and streaming replication is being used (which is the case for EDB Ark). The pgpool community has been alerted to this behavior; please ignore these messages.

Question: How do I stop the Postgres server on a cluster node without triggering a failover process?

To safely stop a Postgres server without triggering failover, you can use either the `service` command or the pg_ctl utility. For more information, see Controlling the Database Server .

Problem: I am attempting to connect to my Advanced Server database with the psql client, and am getting an error

```
(03/23/2012 13:36:53)-> psql --host=192.0.43.10 -p 9999 -U enterprisedb
Password for user enterprisedb: psql: FATAL: database "postgres" does
not exist
```

The psql client expects the default database to be named `postgres`; the edb-psql client expects the default database to be named `edb`. If you attempt to connect to an Advanced Server cluster with the psql client without specifying the name of the database to which it should connect, the client will fail to connect.

You can include the `-d` or `--dbname` flag, followed by the database name when invoking either client to specify the database to which the client should connect.

Question: I'm trying to drop a database from a cluster, but I am getting an error that there are open sessions. There are no clients connected. How can I terminate any leftover backend sessions?

It may be that pgpool is retaining a connection to the database. You can use the `pg_cancel_backend()` or

`pg_terminate_backend()` functions to selectively close connections to the database you wish to drop.

Question: Why do I have to restart pgPool before it will recognize new users that I've added to the database server?

pgPool does not check for new Postgres users. EDB Ark has a periodic update process that updates the user list every 20 seconds; if the update process identifies a new user, it sends a reload signal to the pgPool process. After the reload, pgPool will allow new users to login.

Instead of reloading, simply waiting for 20 seconds between the CREATE USER statement and the CREATEDB statement should solve the problem.

Question: Why are scheduled backups not working?

If you invoke the `pg_start_backup()` function before performing a manual backup your database, you must remember to invoke the `pg_stop_backup()` function when the backup has completed, or EDB Ark scheduled backups will fail.

The EDB Ark Email Notification System

EDB Ark invokes an email notification system that will alert you if your cluster changes or encounters a problem. Email notifications are sent to the address used to log in to the management console.

EDB Ark will send an email:

- When a new cluster is created.
- If a server stops (or is terminated).
- When a replica is added to a cluster.
- When memory is scaled up.
- When failover is invoked on a master or a replica.
- If a backup fails.
- If the password associated with your user account changes.

The `Notification Email` field (on the `User` tab) allows you to change the notification email associated with your user account; for more information, see The User Tab .

2.15 AWS Policies

AWS Security Policy

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

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

"ec2:ModifyVpcAttribute", "ec2:MonitorInstances", "ec2:ReleaseAddress", "ec2:ReplaceNetworkAclAssociation", "ec2:ReplaceNetworkAclEntry", "ec2:ReplaceRoute", "ec2:ReplaceRouteTableAssociation", "ec2:ReportInstanceStatus", "ec2:ResetImageAttribute", "ec2:ResetInstanceAttribute", "ec2:ResetNetworkInterfaceAttribute", "ec2:ResetSnapshotAttribute", "ec2:RevokeSecurityGroup*", "ec2:RunInstances", "ec2:StartInstances", "ec2:UnassignPrivateIpAddresses", "ec2:UnmonitorInstances"], "Resource": "*", "Effect": "Allow", "Sid": "Stmt1407961327680" }, { "Action": ["iam:PassRole"], "Resource": "*", "Effect": "Allow", "Sid": "Stmt1407961362664" }, { "Action": ["s3:CreateBucket", "s3:Get*", "s3:List*"], "Resource": "*", "Effect": "Allow", "Sid": "Stmt1407961630932" }, { "Action": ["s3:Put", "s3:Get*", "s3:DeleteObject*"], "Resource": "arn:aws:s3:::*", "Effect": "Allow", "Sid": "Stmt1407961734627" }, { "Condition": { "StringEquals": { "ec2:ResourceTag/CreatedBy": "EnterpriseDB" } }, "Action": ["ec2:RebootInstances", "ec2:StopInstances", "ec2:TerminateInstances"], "Resource": "*", "Effect": "Allow", "Sid": "Stmt1407961927870" }] }

AWS User Trust Policy

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

{ "Version": "2012-10-17", "Statement": [{ "Sid": "", "Effect": "Allow", "Principal": { "Service": "ec2.amazonaws.com" }, "Action": "sts:AssumeRole" }, { "Sid": "", "Effect": "Allow", "Principal": { "AWS": "arn:aws:iam::747919436152:root" }, "Action": "sts:AssumeRole", "Condition": { "StringEquals": { "sts:ExternalId": "EDB-PPCD-CONSOLE" } } }] }