

# Contents

1 Admin Guide . . . . .	2
2 What's New . . . . .	3
Limitations . . . . .	4
3 Supported Platforms . . . . .	4
4 Ark Architecture Overview . . . . .	5
Using Ark on an AWS Virtual Private Cloud . . . . .	7
Using a NAT Gateway . . . . .	9
5 Registering an Ark Cluster with PEM . . . . .	9
Syncing with the PEM Server . . . . .	11
Monitoring an Ark Cluster . . . . .	14
Registering a PEM Agent . . . . .	17
6 Ark Authentication Models . . . . .	21
Using Provider Authentication on AWS . . . . .	22
Using PostgreSQL Authentication on AWS . . . . .	23
Using Provider Authentication on Azure . . . . .	24
Using PostgreSQL Authentication on Azure . . . . .	25
7.1 Installing the Ark Console . . . . .	26
7.2 Installing the Ark Console on AWS . . . . .	27
Launching the Ark Console Instance on AWS . . . . .	27
Creating the Amazon AWS Service User and Service Role . . . . .	29
Creating the AWS Service User . . . . .	30
Creating the AWS Service Role . . . . .	33
Configuring the Ark Console . . . . .	37
Creating an Amazon Role and Registering an Ark Console User . . . . .	46
7.3 Installing the Ark Console on Azure . . . . .	54
Providing Administrative Access to an Azure User . . . . .	54
Creating a Security Group . . . . .	55
Creating a Storage Account . . . . .	56
Launching the Ark Console Instance . . . . .	58
Configuring the Ark Console . . . . .	68
Connecting to the Administrative Console on an Azure Host . . . . .	78
7 Administrative Features of the EDB Ark Console . . . . .	79
8 Using the Admin Tab . . . . .	81
Using the Console Manager . . . . .	84
Using the Console Manager to Create a Federated Console . . . . .	86
Managing Server Images . . . . .	89
Managing Database Engines . . . . .	93
Adding, Modifying, or Deleting Engine Definitions . . . . .	95
Adding Supporting Components to a Database Engine Definition . . . . .	101
Template Administration . . . . .	104
Red Hat Subscription Management . . . . .	108
Managing Amazon Roles . . . . .	113
User Administration . . . . .	114
User Administration on an Amazon Host . . . . .	117

User Administration on an Azure Host . . . . .	119
Displaying Connected Users . . . . .	120
Managing the Wall Message . . . . .	120
Accessing the Console Logs . . . . .	121
Accessing the Core Usage Logs . . . . .	121
Taking a Manual Backup of the Console . . . . .	122
Editing Installation Properties . . . . .	122
9 Using the Ark DBA Tab . . . . .	122
Accessing User Information . . . . .	124
10 Console Management . . . . .	124
Starting, Stopping or Restarting the Ark Console . . . . .	124
Changing Console Passwords . . . . .	125
Customizing the Console . . . . .	126
Managing Console Logs . . . . .	127
Upgrading the Console . . . . .	127
Updating a PEM Installation . . . . .	129
Recovering From a Console Failure . . . . .	130
Modifying Backup Properties with the EDB Ark Console . . . . .	130
Using the Recover Option . . . . .	132
Manually Recovering from Console Backups . . . . .	133
11 Securing EDB Postgres Ark . . . . .	133
Modifying a Security Group for an Amazon AWS Hosted Console	134
Using ssh to Access a Server . . . . .	135
Using iptables Rules . . . . .	136
Post-Installation Recommendations . . . . .	137
12 Creating a Statically Provisioned Image . . . . .	137
13 EDB Postgres Ark Notifications . . . . .	138
14 EDB Postgres Ark Resources . . . . .	139
15 'AWS IAM Role Permission Policy' . . . . .	139
16 Amazon IAM Role Trust Relationship . . . . .	142
17 Amazon Service User Security Policy . . . . .	142
18 Conclusion . . . . .	143

---

## 1 Admin Guide

### 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.

ark\_whats\_new ark\_supported\_platforms ark\_architecture\_overview  
ark\_registering\_with\_pem ark\_authentication\_models ark\_installing\_console  
ark\_admin\_features ark\_admin\_tab ark\_dba\_tab ark\_console\_management  
ark\_securing\_ark ark\_statically\_provisioned ark\_notifications ark\_resources  
ark\_statically\_provisioned ark\_aws\_iam\_role\_permission\_policy ark\_aws\_iam\_role\_trust\_relationship  
ark\_aws\_service\_user\_security\_policy conclusion

---

## 2 What's New

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

### New Features

- 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-3/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`.

---

## **3 Supported Platforms**

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

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

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

- RHEL 7.x
  - CentOS 7.x
- 

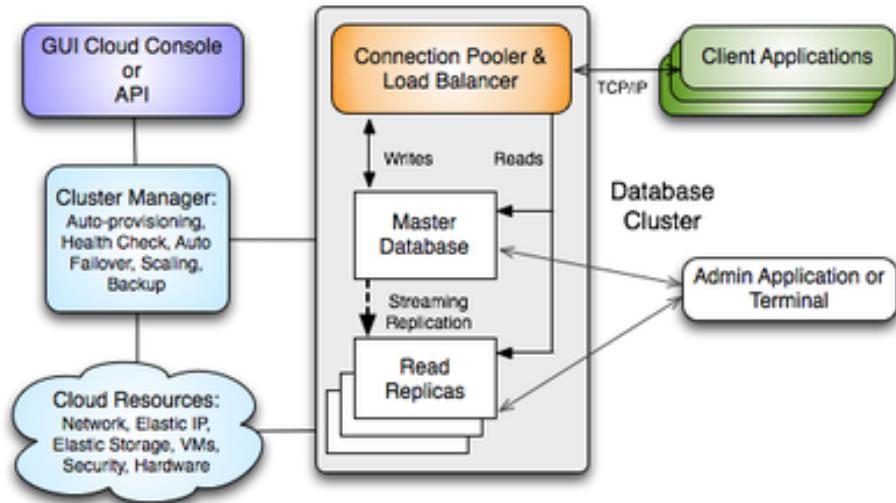
## 4 Ark Architecture Overview

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.

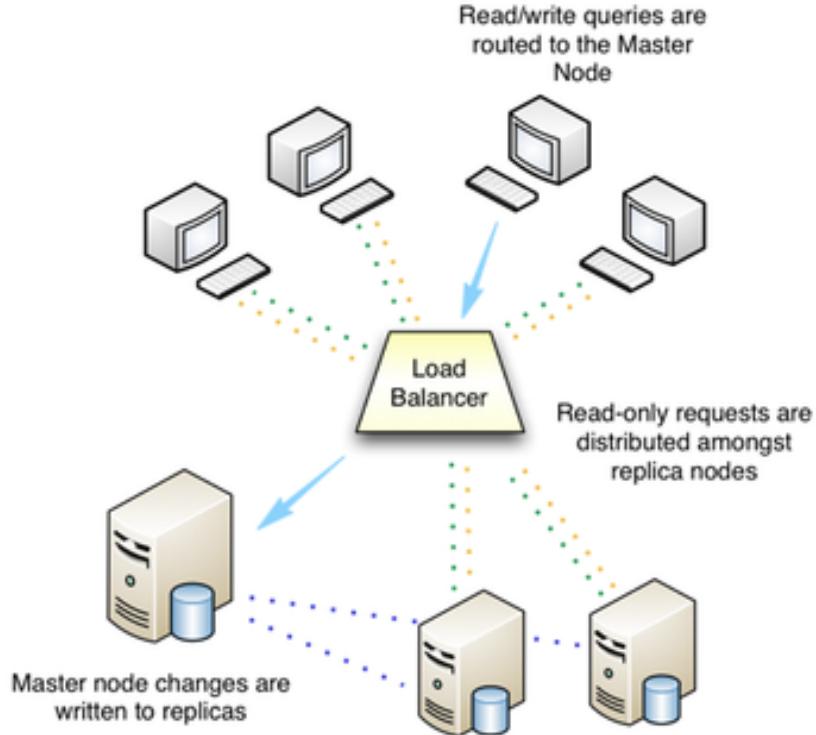
When you create a new cluster (a group of replicated database servers), EDB Ark initializes one or more Postgres instances (virtual machines) according to your specifications. EDB Ark uses Postgres streaming replication to synchronize replicas in the cluster, and pgpool-II to implement load balancing 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.

Create a New Server Cluster

Step 1 Step 2 Step 3 Step 4

Provide the details for your cluster

Cluster Name	<input type="text"/>
Engine Version	PostgreSQL 11 64bit on CentOS / RHEL 7
Server Class	t3.micro
vCPU: 2, Mem: 1GB	
<input type="checkbox"/> Use Private IP addresses	
VPC	New VPC
Number of nodes	1
Storage GB	1
<input type="checkbox"/> Encrypted	
<input type="checkbox"/> EBS Optimized	
IOPS	0
<input type="checkbox"/> Perform OS and Software update?	
Master User	postgres
Master Password	postgres
Notification Email	susan.douglas@enterprisedb.com

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`.

---

## 5 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.

Use the following properties to configure integration with a PEM server:

PEM Server Mode	REMOTE
PEM Server IP Address	
PEM Server DB Port	
PEM Server API Port	
PEM Server Username	
PEM Server Password	
PEM Sync Mode	ENABLED
PEM Synchronization Interval	10

During deployment, you will be prompted to use the **PEM Server Mode** dropdown 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**

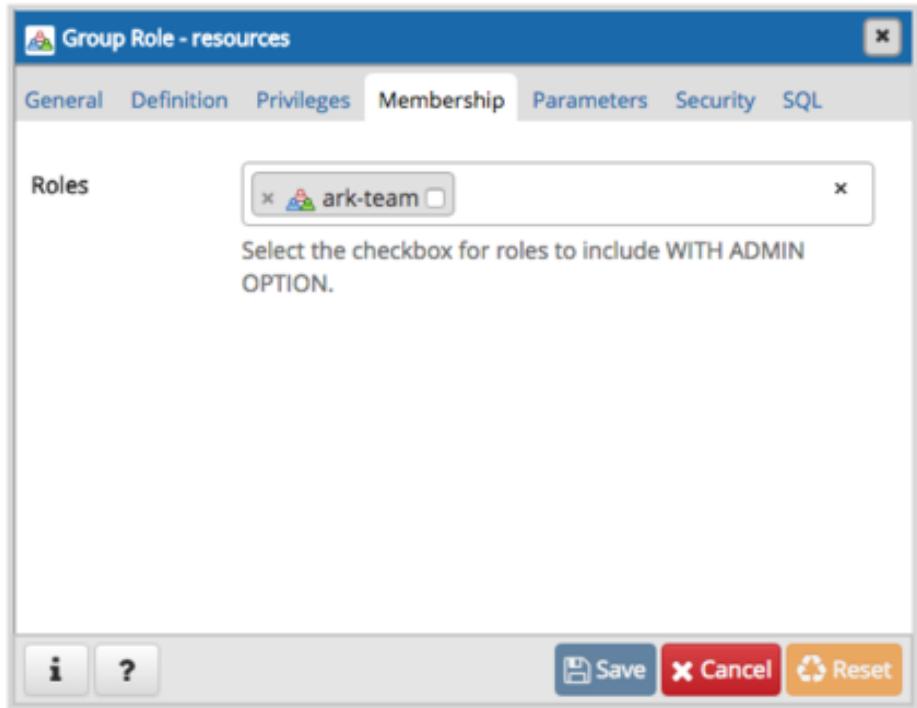
When you register with PEM during console deployment, the *PEM Sync Mode* field will be enabled. Use the drop-down listbox in the PEM Sync Mode field to specify your preference for synchronizing with the PEM server.

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.

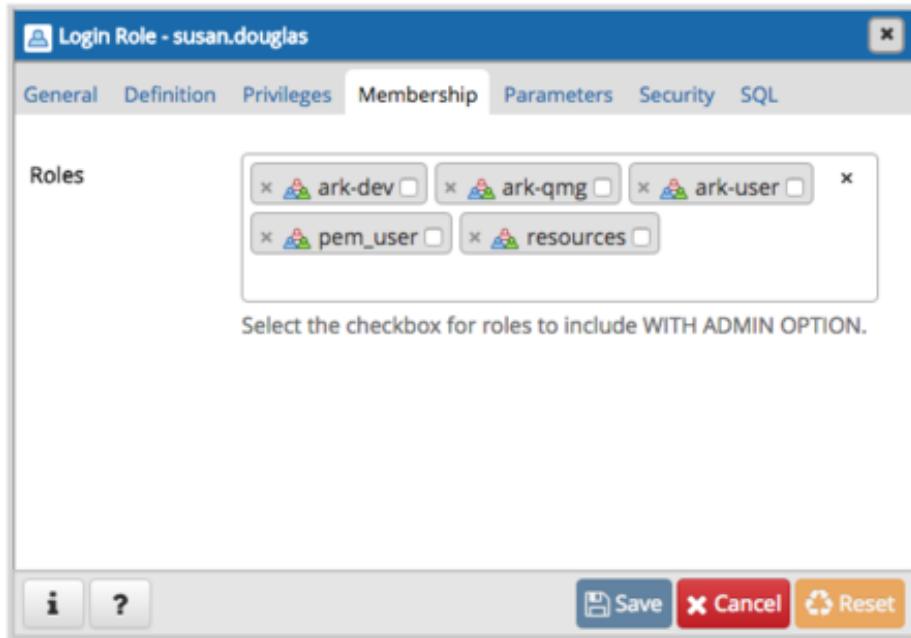
If you select **ENABLED**:

- 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.



- 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.

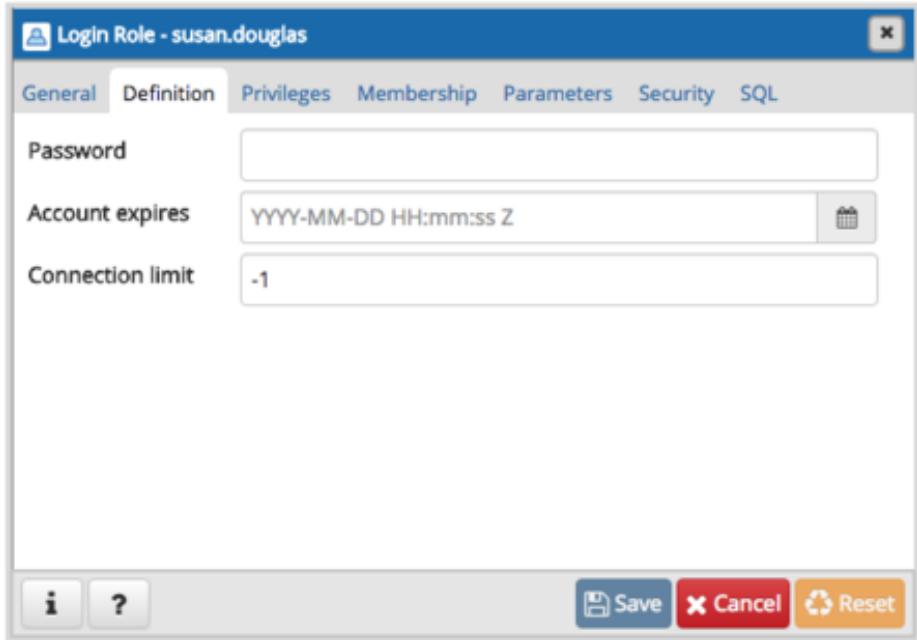
Each PEM user account that corresponds with a registered Ark user will belong to the `ark-user` role. The PEM user account will also have membership in any `ark-team` roles that have been created to correspond to the project, role, or groups that the Ark user has permission to access.



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.



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.

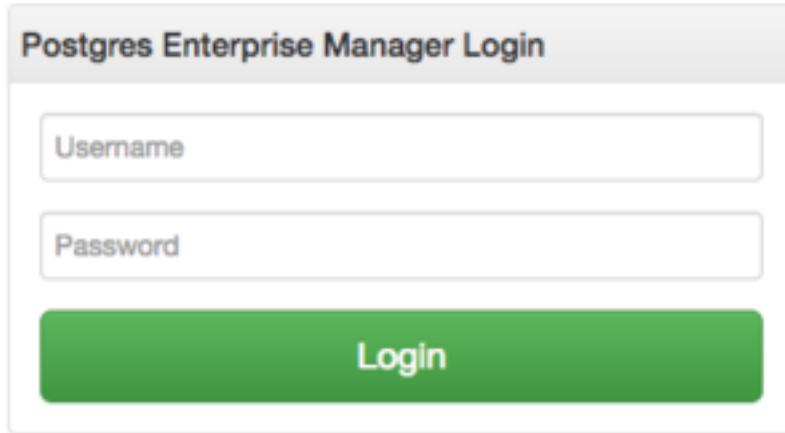
Monitoring an Ark Cluster

#### 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](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](https://<address_of_remote_pem_server>:<port>/pem)

When prompted, provide the PEM credentials to connect.



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.

The screenshot shows the PEM web interface. On the left, a tree control labeled 'Browser' shows a cluster named 'Ark-Cluster\_sales (6)' with several nodes under it. The main area is titled 'Global Overview' with a timestamp of '2018-05-29 16:15:43' and 'Number of Alerts: 10'. It features a bar chart titled 'Status' comparing 'Agents Up' and 'Servers Up' against 'Agents Down' and 'Servers Down'. Below this are two tables: 'Agent Status' and 'Postgres Server Status', both listing various nodes with their status, alerts, version, processes, threads, CPU utilization, memory utilization, and swap usage.

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

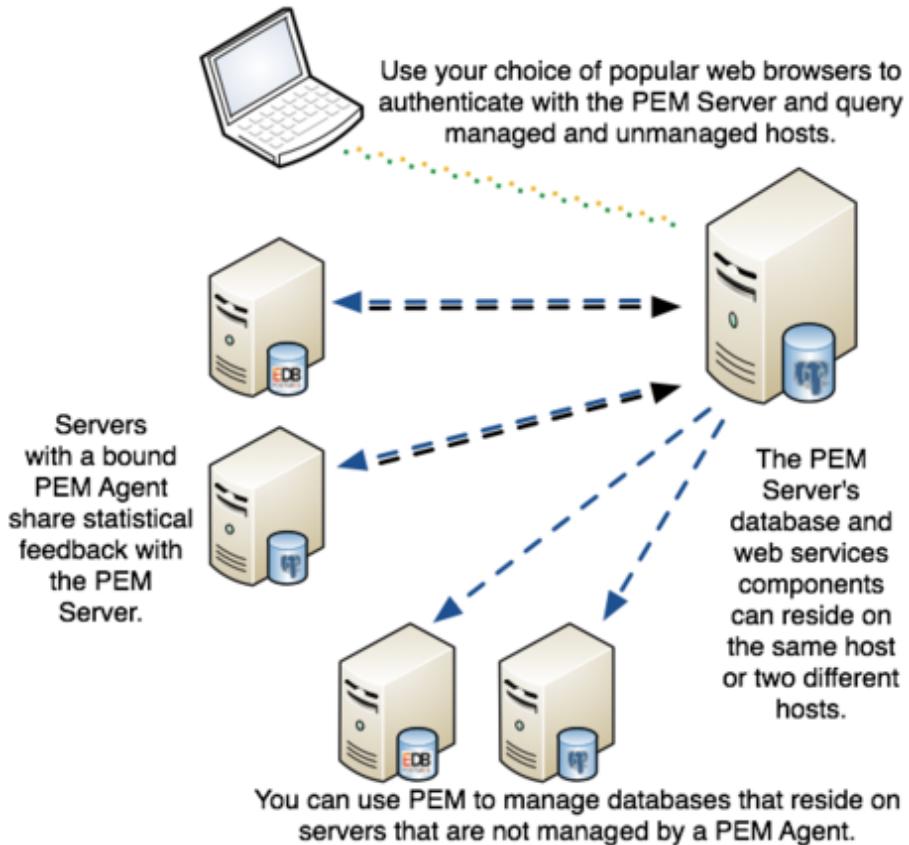
If your Ark clusters are provisioned with private IP addresses, they may not be reachable from the PEM server. If this is the case, you will not be able to use PEM to remotely browse the database server. PEM agents running on the Ark cluster nodes will be able to report status to the PEM Server.

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

## 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.



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](http://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.

```

/var/lib/ppas/9.5/data
[root@sales-2d6 data]# vi pg_hba.conf

Hostssl all all 192.168.2.248/32 md5
hostssl all all 192.168.2.248/32 cert

# PostgreSQL Client Authentication Configuration File
# =====
#
# Refer to the "Client Authentication" section in the PostgreSQL
# documentation for a complete description of this file. A short
# synopsis follows.
#
# This file controls: which hosts are allowed to connect, how clients
# are authenticated, which PostgreSQL user names they can use, which
# databases they can access. Records take one of these forms:
#
# local      DATABASE  USER  METHOD  [OPTIONS]
# host       DATABASE  USER  CIDR-ADDRESS METHOD  [OPTIONS]
# hostssl    DATABASE  USER  CIDR-ADDRESS METHOD  [OPTIONS]
# hostnoss  DATABASE  USER  CIDR-ADDRESS METHOD  [OPTIONS]
#

```

### **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>
```

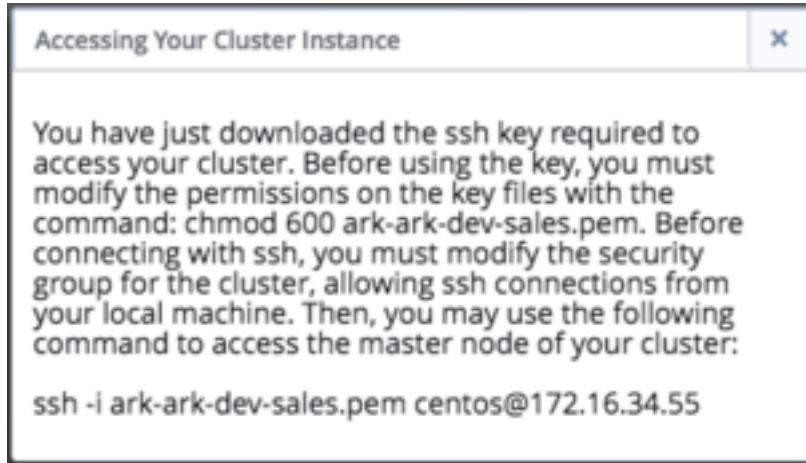
Where <x> specifies the version. For example, the following command restarts a PostgreSQL 10 service:

```
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.



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>
```

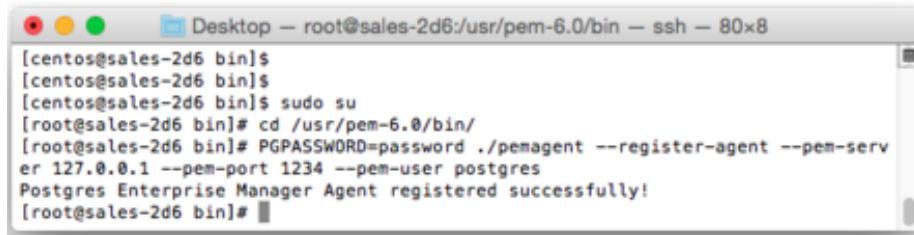
Where:

<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.



A screenshot of a terminal window titled "Desktop - root@sales-2d6:/usr/pem-6.0/bin - ssh - 80x8". The terminal shows the following command sequence:

```
[centos@sales-2d6 bin]$  
[centos@sales-2d6 bin]$  
[centos@sales-2d6 bin]$ sudo su  
[root@sales-2d6 bin]# cd /usr/pem-6.0/bin/  
[root@sales-2d6 bin]# PGPASSWORD=password ./pemagent --register-agent --pem-server 127.0.0.1 --pem-port 1234 --pem-user postgres  
Postgres Enterprise Manager Agent registered successfully!  
[root@sales-2d6 bin]#
```

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 .

---

## 6 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:

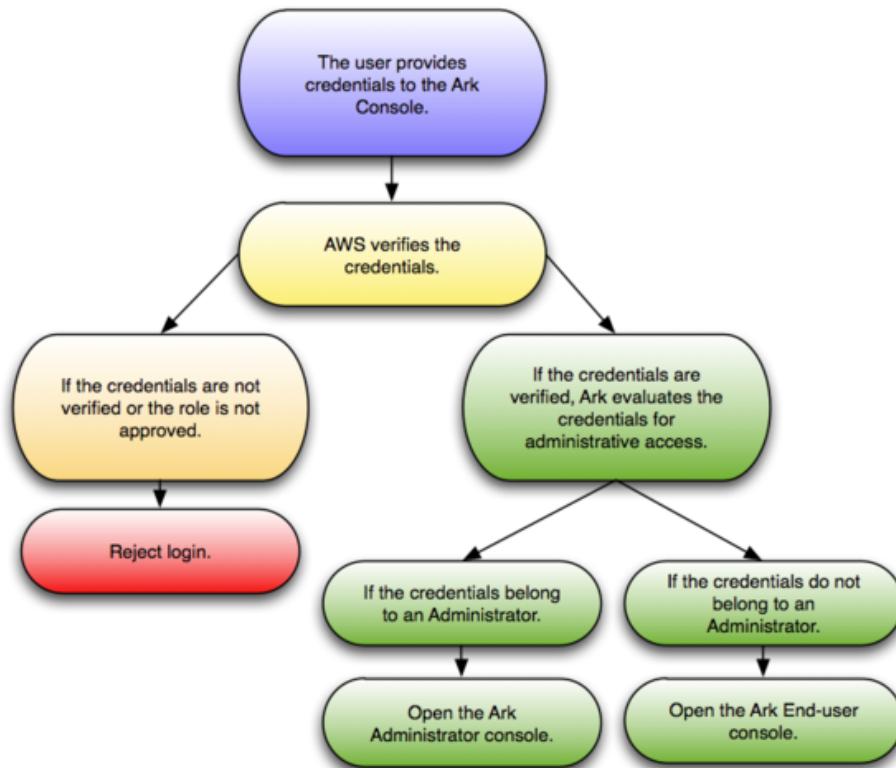
<**user\_name**> specifies the name of the Ark user. <**password**> specifies the password associated with the user name.

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.

#### 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.

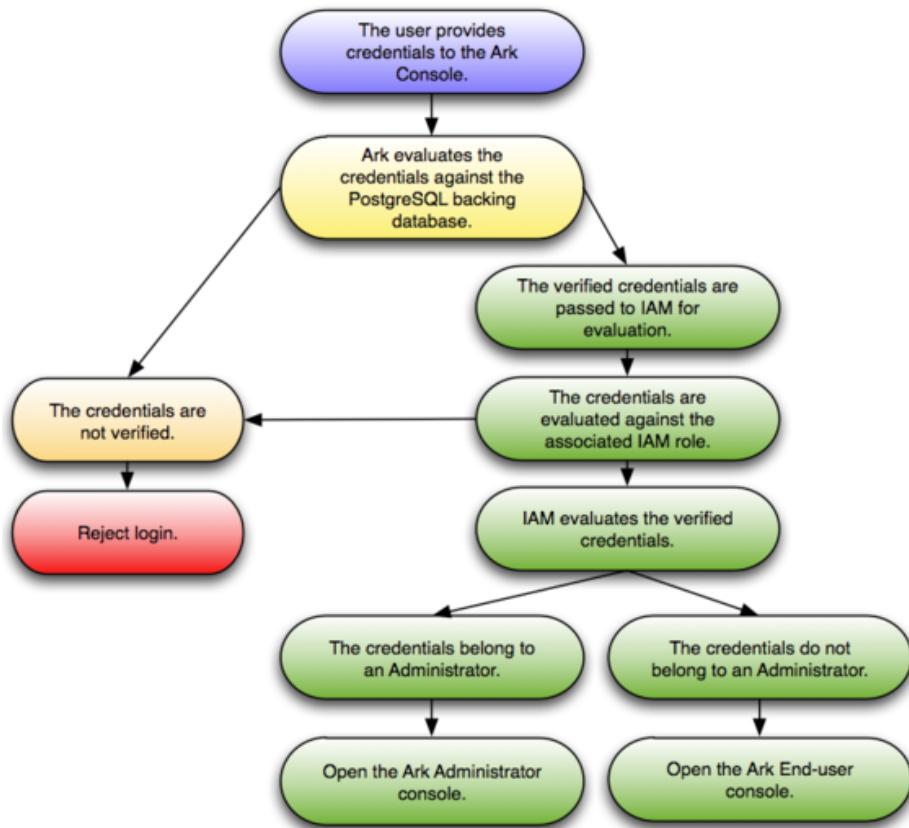


When the user provides credentials to the Ark console, the credentials are passed to Amazon for verification. If the credentials are successfully verified, the role is evaluated to determine if the user should have access to the Administrator console or the end-user console.

#### PostgreSQL Authentication on AWS

##### **Using PostgreSQL Authentication on AWS**

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

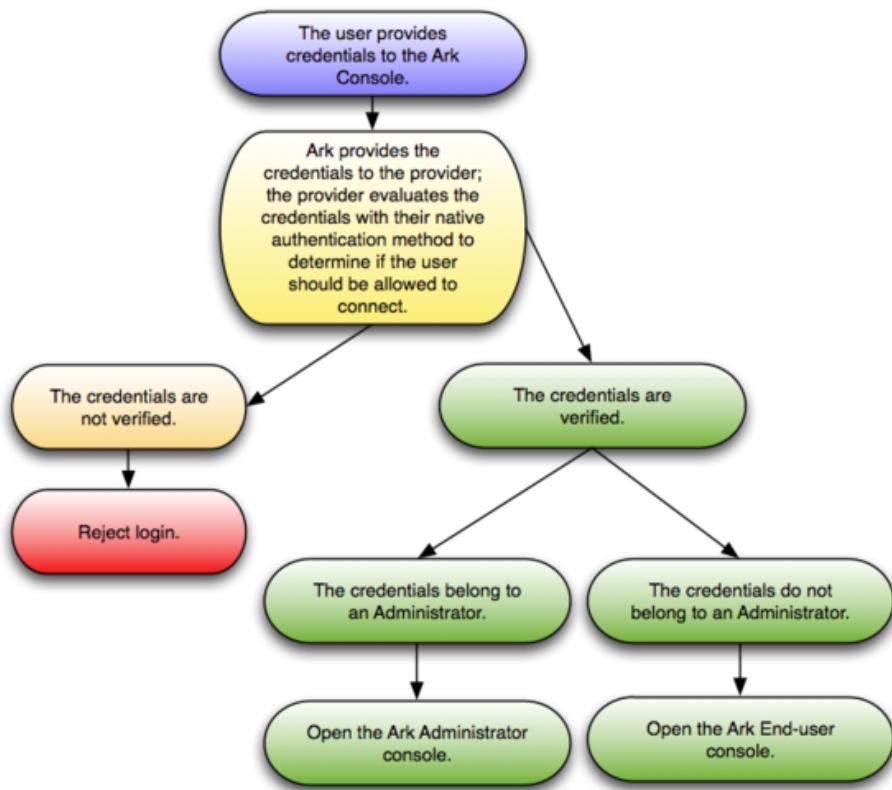


An Ark Administrative user must use a client application (such as psql or PEM) to add each user to the Ark backing database, and then use the User Administration table to register Ark users. The user name and associated password specified in the Ark backing database must match the credentials specified 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.

#### Provider Authentication on Azure

##### **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.

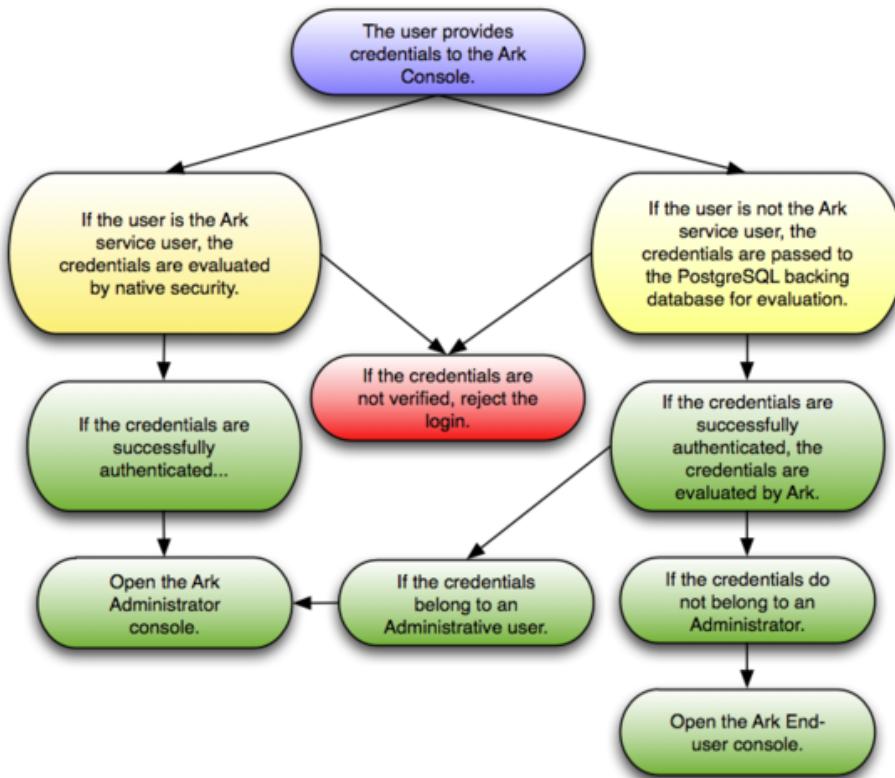


When the user provides credentials to the Ark console, the credentials are passed to the provider for verification. If the credentials are successfully verified, the role is evaluated to determine if the user should have access to the Administrator console or the end-user console.

#### PostgreSQL Authentication on Azure

##### **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.



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.

---

## 7.1 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.

ark\_installing\_console\_aws ark\_installing\_console\_azure

---

## 7.2 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 on AWS

### Launching the Ark Console Instance on AWS

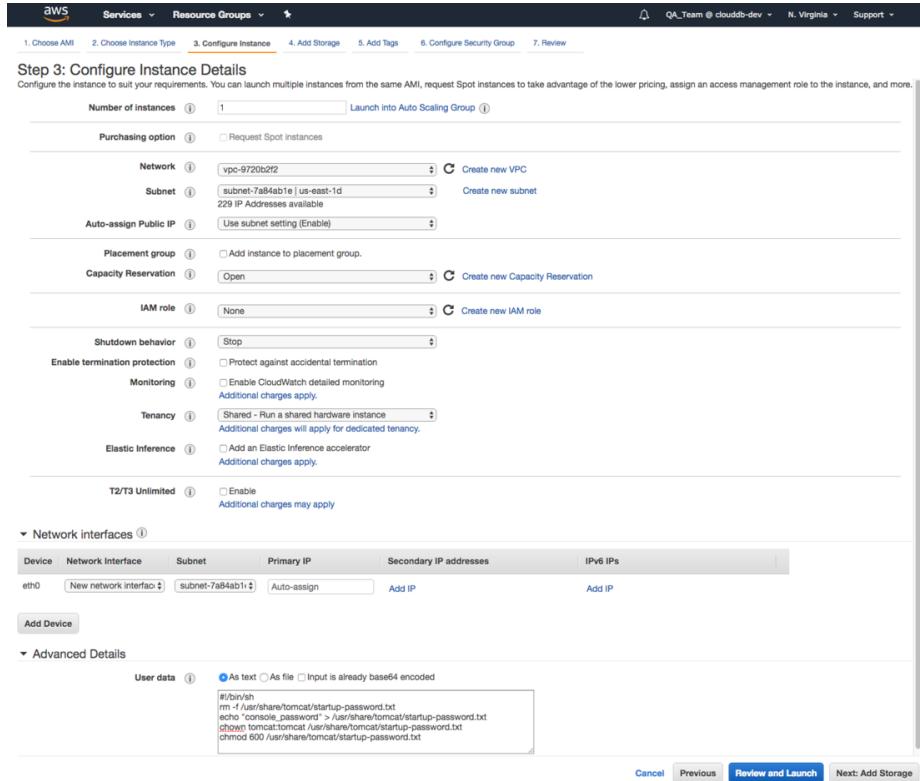
Before launching an AMI into an Amazon VPC, you must ensure that the VPC has access to an Internet Gateway. If your VPC does not have access to an Internet Gateway, you can use the Amazon management console to create an Internet Gateway and associate it with your VPC.

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.



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 AWS Service User and Role

**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

## 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.

The screenshot shows the 'Add user' dialog with four steps numbered 1 to 4. Step 1 is active, titled 'Set user details'. It includes a 'User name\*' field containing 'acctg-svc-user', an 'Add another user' link, and a note about adding multiple users with the same access type. Step 2, 'Select AWS access type', is shown below. It has two options: 'Programmatic access' (selected with a checked checkbox) and 'AWS Management Console access'. A note states that programmatic access provides an access key ID and secret access key for the AWS API, CLI, SDK, and other development tools. The 'AWS Management Console access' option enables a password for the AWS Management Console. At the bottom are 'Cancel' and 'Next: Permissions' buttons. A note at the bottom left indicates that the user name is required.

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.

Create policy

A policy defines the AWS permissions that you can assign to a user, group, or role. You can create and edit a policy in the visual editor and using JSON. [Learn more](#)

[Visual editor](#) [JSON](#) [Import managed policy](#)

```

3.   "Statement": [
4.     {
5.       "Sid": "Stmt1389628412000",
6.       "Effect": "Allow",
7.       "Action": [
8.         "sts:GetFederationToken",
9.         "sts:AssumeRole"
10.      ],
11.      "Resource": [
12.        "*"
13.      ]
14.    }
15.  ]
16. }
17.

```

[Cancel](#) [Review policy](#)

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

Create policy

[1](#) [2](#)

Review policy

Name\*  Use alphanumeric and '+,-,@-\_.' characters. Maximum 128 characters.

Description  Maximum 1000 characters. Use alphanumeric and '+,-,@-\_.' characters.

Summary

Q Filter		
Service ▾	Access level	Resource
Allow (1 of 136 services) <a href="#">Show remaining 135</a>	Limited: Read, Write	All resources
STS		

\* Required [Cancel](#) [Previous](#) [Create policy](#)

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

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.

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.

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

#### 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.

Create role

Select type of trusted entity

1 2 3

 AWS service EC2, Lambda and others	 Another AWS account Belonging to you or 3rd party	 Web identity Cognito or any OpenID provider	 SAML 2.0 federation Your corporate directory
---	--	--	---

Allows AWS services to perform actions on your behalf. [Learn more](#)

Choose the service that will use this role

**EC2**  
Allows EC2 instances to call AWS services on your behalf.

**Lambda**  
Allows Lambda functions to call AWS services on your behalf.

API Gateway	Config	Elastic Container Service	Lex	SWF
AppSync	DMS	Elastic Transcoder	Machine Learning	SageMaker
Application Auto Scaling	Data Pipeline	Elastic Load Balancing	MediaConvert	Service Catalog
Auto Scaling	DeepLens	Glue	OpsWorks	Step Functions
Batch	Directory Service	Greengrass	RDS	Storage Gateway
CloudFormation	DynamoDB	GuardDuty	Redshift	
CloudHSM	EC2	Inspector	Rekognition	
CloudWatch Events	EMR	IoT	S3	
CodeBuild	ElastiCache	Kinesis	SMS	
CodeDeploy	Elastic Beanstalk	Lambda	SNS	

\* Required

Cancel **Next: Permissions**

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

**Create role**

**Attach permissions policies**

Choose one or more policies to attach to your new role.

Policy name	Attachments	Description
acctg-svc-user-policy	1	This is the service user policy for the accounting department.
AdministratorAccess	5	Provides full access to AWS services and resources.
AlexaForBusinessDeviceSetup	0	Provide device setup access to AlexaForBusiness services.
AlexaForBusinessFullAccess	0	Grants full access to AlexaForBusiness resources and accounts.
AlexaForBusinessGatewayExecution	0	Provide gateway execution access to AlexaForBusiness services.
AlexaForBusinessReadOnlyAccess	0	Provide read only access to AlexaForBusiness services.
AmazonAPIGatewayAdministrator	0	Provides full access to create/edit/delete APIs in Amazon API Gateway.
AmazonAPIGatewayInvokeFullAccess	0	Provides full access to invoke APIs in Amazon API Gateway.
AmazonAPIGatewayPushToCloudWatchLogs	0	Allows API Gateway to push logs to user's account.
AmazonAppStreamFullAccess	0	Provides full access to Amazon AppStream via the AWS Management Console.

\* Required      Cancel      Previous      **Next: Review**

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

**Create role**

**Review**

Provide the required information below and review this role before you create it.

Role name*	acctg_service_role
Use alphanumeric and '+,-, @-' characters. Maximum 64 characters.	
Role description	This is the service role for the accounting department.
Maximum 1000 characters. Use alphanumeric and '+,-, @-' characters.	
Trusted entities	AWS service: ec2.amazonaws.com
<b>Policies</b>	

\* Required      Cancel      Previous      **Create role**

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.

Roles > acctg\_service\_role

**Summary**

**Delete role**

Role ARN	arn:aws:iam::325753300792:role/acctg_service_role
Role description	This is the service role for the accounting department. <a href="#">Edit</a>
Instance Profile ARNs	arn:aws:iam::325753300792:instance-profile/acctg_service_role
Path	/
Creation time	2018-04-11 12:22 EDT
Maximum CLI/API session duration	1 hour <a href="#">Edit</a>

**Permissions** [Trust relationships](#) [Access Advisor](#) [Revoke sessions](#)

**Get started with permissions**  
This role doesn't have any permissions yet. Get started by attaching one or more policies to this role. [Learn more](#)

**Attach policy**

**Add inline policy**

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.

**Create policy**

1 [1](#) 2 [2](#)

A policy defines the AWS permissions that you can assign to a user, group, or role. You can create and edit a policy in the visual editor and using JSON. [Learn more](#)

[Visual editor](#) **JSON** [Import managed policy](#)

```

1 - {
2 "Version": "2012-10-17",
3 "Statement": [ {
4 "Action": [
5 "ec2:AllocateAddress",
6 "ec2:AssignPrivateIpAddresses",
7 "ec2:Associate*",
8 "ec2:Attach*",
9 "ec2:AuthorizeSecurityGroup*",
10 "ec2:Copy*",
11 "ec2:Create*",
12 "ec2:DeleteInternetGateway",
13 "ec2:DeleteNetworkAcl",
14 "ec2:DeleteNetworkAclEntry",
15 "ec2:DeleteNetworkInterface"
]
}

```

**Cancel** **Review policy**

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.

## Edit Trust Relationship

You can customize trust relationships by editing the following access control policy document.

### Policy Document

```
1- {
2-   "Version": "2012-10-17",
3-   "Statement": [
4-     {
5-       "Sid": "",
6-       "Effect": "Allow",
7-       "Principal": {
8-         "Service": "ec2.amazonaws.com"
9-       },
10-      "Action": "sts:AssumeRole"
11-    },
12-    {
13-      "Sid": "",
14-      "Effect": "Allow",
15-      "Principal": {
16-        "AWS": "arn:aws:iam::305753120797:root"
17-      },
18-      "Action": "sts:AssumeRole",
19-      "Condition": {
20-        "StringEquals": {
21-          "sts:ExternalId": "EDB-ARK-SERVICE"
22-        }
23-      }
24-    }
25-  ]
26- }
```

[Cancel](#) [Update Trust Policy](#)

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.

**Role ARN** arn:aws:iam::325753300792:role/acctg\_service\_role

**Role description** This is the service role for the accounting department. | Edit

**Instance Profile ARNs** arn:aws:iam::325753300792:instance-profile/acctg\_service\_role

**Path** /

**Creation time** 2018-04-11 12:22 EDT

**Maximum CLI/API session duration** 1 hour Edit

**Permissions** **Trust relationships** **Access Advisor** **Revoke sessions**

You can view the trusted entities that can assume the role and the access conditions for the role. [Show policy document](#)

**Edit trust relationship**

Trusted entities			Conditions		
Condition	Key	Value	Condition	Key	Value
StringEquals	sts:ExternalId	4a44daac-2e92-42b3-86287844a77b			

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

### Configuring the Ark Console

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



Please enter the password specified at launch time. If you did not specify your own password, you can find a generated one by logging into the console server using SSH and reading the file /usr/share/tomcat/startup-password.txt.

Password

Deploy Console

Recover From Backup

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

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



EDB Ark

Use the following fields to set Ark console properties.

These properties are specific to the Amazon EC2 provider:

AWS Access Key

AWS Secret Key

Service Account Role ARN

Service Account External ID

Service Account ID

Service Account Password

Enable Self Registration

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.

Provide general server properties in the following section:

Console DNS Name	<input type="text"/>
Contact Email Address	<input type="text"/>
Email From Address	<input type="text"/>
Notification Email	<input type="text"/>
Cc From Address	<input type="text"/>
API Timeout	<input type="text"/>
WAL Archive Container	<input type="text"/>
Dashboard Docs URL	<input type="text"/>
Dashboard Hot Topics URL	<input type="text"/>
Enable Console Manager	<input type="text"/>
Enable Postgres Authentication	<input type="text"/>
Template Restrict New Users	<input type="text"/>
Cluster Event Retention Limit	<input type="text"/>

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.

Use the following properties to configure integration with a PEM server:

PEM Server Mode	<input type="button" value="REMOTE"/>
PEM Server Address	<input type="text"/>
IP address:	
PEM Server DB Port	<input type="text"/>
PEM Server API Port	<input type="text"/>
PEM Server Username	<input type="text"/>
PEM Server Password	<input type="text"/>
PEM Sync Mode	<input type="button" value="DISABLED"/>

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.

### saml Configuring Ark for SAML

Use the following properties to configure the SAML service provider:

SAML Auth Enabled	<input type="text" value="true"/>
SP Entity ID	<input type="text"/>
SP Consumer Service URL	<input type="text"/>
SP Consumer Service Binding	<input type="text"/>
SP Logout Service URL	<input type="text"/>
SP Logout Service Binding	<input type="text"/>
SP Name ID Format	<input type="text"/>
SP Certificate	<input type="text"/>
SP Private Key	<input type="text"/>

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.

IDP Entity ID	<input type="text"/>
IDP Sign On URL	<input type="text"/>
IDP Sign On Service Binding	<input type="text"/>
IDP Logout Service URL	<input type="text"/>
IDP Logout Service Response URL	<input type="text"/>
IDP Single Logout Service Binding	<input type="text"/>
IDP Certificate	<input type="text"/>

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.

Encrypted Name ID	<input type="text"/>
Auth Request Signed	<input type="text"/>
Logout Request	<input type="text"/>
Logout Response Signed	<input type="text"/>
Sign messages	<input type="text"/>
Sign assertions	<input type="text"/>
Sign Metadata	<input type="text"/>
Encrypt Assertions	<input type="text"/>
Encrypt Name ID	<input type="text"/>
Authentication Context	<input type="text"/>
Auth Comparison Parameters	<input type="text"/>
XML validation	<input type="text"/>
Signature Algorithm	<input type="text"/>

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`

Organization Name	<input type="text"/>
Display Name	<input type="text"/>
Organization URL	<input type="text"/>
Organization Language	<input type="text"/>
Technical Contact Name	<input type="text"/>
Technical Email Address	<input type="text"/>
Support Contact Name	<input type="text"/>
SAML Support Email Address	<input type="text"/>

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.

Use the following properties to enable console backup storage:

Storage Bucket	<input type="text"/>
Console Backup Folder	<input type="text"/>

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.

Use the following properties to change password for DB user:

DB User New Password	<input type="text"/>
DB User Confirm Password	<input type="text"/>

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**).

Specify a timezone for the server:

Timezone	<input type="text"/> <input type="button" value="▼"/>
----------	---

Click Save to preserve your edits, validate the properties with the service provider, and configure and deploy the Ark console.

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 AWS Role and Registering an Ark User

#### 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.

**Create role**

1    2    3

Select type of trusted entity

**AWS service** EC2, Lambda and others    **Another AWS account** Belonging to you or 3rd party    **Web identity** Cognito or any OpenID provider    **SAML 2.0 federation** Your corporate directory

Allows AWS services to perform actions on your behalf. [Learn more](#)

Choose the service that will use this role

**EC2** Allows EC2 instances to call AWS services on your behalf.

**Lambda** Allows Lambda functions to call AWS services on your behalf.

API Gateway	Config	Elastic Container Service	Lex	SWF
AppSync	DMS	Elastic Transcoder	Machine Learning	SageMaker
Application Auto Scaling	Data Pipeline	Elastic Load Balancing	MediaConvert	Service Catalog
Auto Scaling	DeepLens	Glue	OpsWorks	Step Functions
Batch	Directory Service	Greengrass	RDS	Storage Gateway
CloudFormation	DynamoDB	GuardDuty	Redshift	
CloudHSM	EC2	Inspector	Rekognition	
CloudWatch Events	EMR	IoT	S3	
CodeBuild	ElastiCache	Kinesis	SMS	
CodeDeploy	Elastic Beanstalk	Lambda	SNS	

\* Required    [Cancel](#)    [Next: Permissions](#)

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

## Create role

1 2 3

### Attach permissions policies

Choose one or more policies to attach to your new role.

[Create policy](#) [Refresh](#)

	Policy name	Attachments	Description
<input type="checkbox"/>	acctg-policy	0	Use this policy for acctg related activity.
<input type="checkbox"/>	acctg-service-user-policy	0	This is the service user security policy.
<input type="checkbox"/>	acctg-svc-user-policy	1	This is the service user policy for the accounting department.
<input type="checkbox"/>	AdministratorAccess	5	Provides full access to AWS services and resources.
<input type="checkbox"/>	AlexaForBusinessDeviceSetup	0	Provide device setup access to AlexaForBusiness services.
<input type="checkbox"/>	AlexaForBusinessFullAccess	0	Grants full access to AlexaForBusiness resources and accounts.
<input type="checkbox"/>	AlexaForBusinessGatewayExecution	0	Provide gateway execution access to AlexaForBusiness services.
<input type="checkbox"/>	AlexaForBusinessReadOnlyAccess	0	Provide read only access to AlexaForBusiness services.
<input type="checkbox"/>	AmazonAPIGatewayAdministrator	0	Provides full access to create/edit/delete APIs in Amazon API Gateway.
<input type="checkbox"/>	AmazonAPIGatewayInvokeFullAccess	0	Provides full access to invoke APIs in Amazon API Gateway.
<input type="checkbox"/>	AmazonAPIGatewayPushToCloudWatchLogs	0	Allows API Gateway to push logs to user's account.
<input type="checkbox"/>	AmazonAppStreamFullAccess	0	Provides full access to Amazon AppStream via the AWS Management Console.

\* Required

[Cancel](#) [Previous](#) [Next: Review](#)

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

## Create role

1 2 3

### Review

Provide the required information below and review this role before you create it.

Role name\*

acctg\_admin

Use alphanumeric and '+,-, @,-' characters. Maximum 64 characters.

Role description

This is the administrative account for the accounting department.

Maximum 1000 characters. Use alphanumeric and '+,-, @,-' characters.

Trusted entities AWS service: ec2.amazonaws.com

Policies

\* Required

[Cancel](#) [Previous](#) [Create role](#)

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.

Roles > acctg\_admin

**Summary**

**Delete role**

Role ARN	arn:aws:iam::325753300792:role/acctg_admin
Role description	This is the administrative account for the accounting department. <a href="#">Edit</a>
Instance Profile ARNs	arn:aws:iam::325753300792:instance-profile/acctg_admin
Path	/
Creation time	2018-04-11 13:20 EDT
Maximum CLI/API session duration	1 hour <a href="#">Edit</a>

**Permissions** **Trust relationships** **Access Advisor** **Revoke sessions**

**Get started with permissions**  
This role doesn't have any permissions yet. Get started by attaching one or more policies to this role. [Learn more](#)

**Attach policy**

**Add inline policy**

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.

**Create policy**

1 2

A policy defines the AWS permissions that you can assign to a user, group, or role. You can create and edit a policy in the visual editor and using JSON. [Learn more](#)

**Visual editor** **JSON** **Import managed policy**

```

1  {
2   "Version": "2012-10-17",
3   "Statement": [
4     {
5       "Action": [
6         "ec2:AllocateAddress",
7         "ec2:AssignPrivateIpAddresses",
8         "ec2:Associate*",
9         "ec2:Attach*",
10        "ec2:AuthorizeSecurityGroup*",
11        "ec2:Copy*",
12        "ec2>Create*",
13        "ec2>DeleteInternetGateway",
14        "ec2>DeleteNetworkAcl",
15        "ec2>DeleteNetworkAclEntry",
16        "ec2>DeleteNetworkInterface"
17      ],
18      "Effect": "Allow",
19      "Resource": "*"
20    }
21  ]
22 }
```

**Cancel** **Review 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.

## Edit Trust Relationship

You can customize trust relationships by editing the following access control policy document.

### Policy Document

```
1 - {
2   "Version": "2012-10-17",
3   "Statement": [
4     {
5       "Sid": "",
6       "Effect": "Allow",
7       "Principal": {
8         "Service": "ec2.amazonaws.com"
9       },
10      "Action": "sts:AssumeRole"
11    },
12    {
13      "Sid": "",
14      "Effect": "Allow",
15      "Principal": {
16        "AWS": "arn:aws:iam::325753300792:root"
17      },
18      "Action": "sts:AssumeRole",
19      "Condition": {
20        "StringEquals": {
21          "sts:ExternalId": "09fc4a59-dc7e-451e-83a4-4725a5488e61"
22        }
23      }
24    }
25  ]
26 }
```

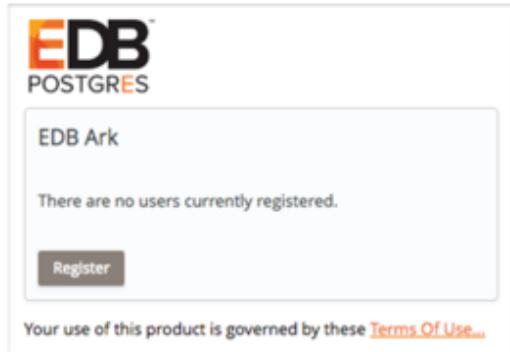
[Cancel](#) [Update Trust Policy](#)

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.



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

A modal dialog titled "New User Registration". It has two tabs at the top: "Step 1" (which is active) and "Step 2". The "User Details" tab is selected. The form contains fields for First Name, Last Name, Password, Verify Password, Email, Cloud Provider (with a dropdown menu showing "amazon ZON EC2"), and Company Name. At the bottom are "Cancel" and "Next" buttons.

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.

The screenshot shows the 'Summary' tab of the IAM role configuration for 'acctg\_admin'. Key details include:

- Role ARN:** arn:aws:iam::325753300792:role/acctg\_admin
- Role description:** This is the administrative account for the accounting department. | Edit
- Instance Profile ARNs:** arn:aws:iam::325753300792:instance-profile/acctg\_admin
- Path:** /
- Creation time:** 2018-04-11 13:20 EDT
- Maximum CLI/API session duration:** 1 hour | Edit
- Give this link to users who can switch roles in the console:** [https://signin.aws.amazon.com/switchrole?roleName=acctg\\_admin&account=cloud9-dev](https://signin.aws.amazon.com/switchrole?roleName=acctg_admin&account=cloud9-dev)

Below the summary, there are tabs for **Permissions**, **Trust relationships**, **Access Advisor**, and **Revoke sessions**. The **Trust relationships** tab is selected, showing:

- Edit trust relationship** button
- Trusted entities:** The identity provider(s) ec2.amazonaws.com and The account 325753300792
- Conditions:** A table showing a single condition: StringEquals st:ExternalId 09fc4a59-dc7e-451e-83a4-4725a54886e1

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

New User Registration

Step 1 Step 2

Enter your Amazon EC2 Security Credentials

Role Arn

Your External ID

[Previous](#) [Cancel](#) [Finish](#)

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.

**EDB POSTGRES**

EDB Ark

Please Log In

Email

Password

[Forgot your password?](#)

[Log In](#)

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

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.

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
- 

### 7.3 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 <#provide\_access\_azure>`\_ to access the Ark Administrator's console. Create an Azure network security group .

Create an `Azure storage account <#azure\_storage\_account>`\_ . Launch a VM Image that contains the Ark console.

Configure the `Ark console <#azure\_configure\_ark\_console>`\_ . Register an Ark Administrative user .

Providing Admin access to an Azure 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 on Azure

##### 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	0.0.0.0/0
All ICMP	Ingress			CIDR	
SSH	Ingress			CIDR	
HTTP				CIDR	
HTTPS				CIDR	
Custom TCP	Ingress	6666		CIDR	

Rule Type	Direction	Port	Remote	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.

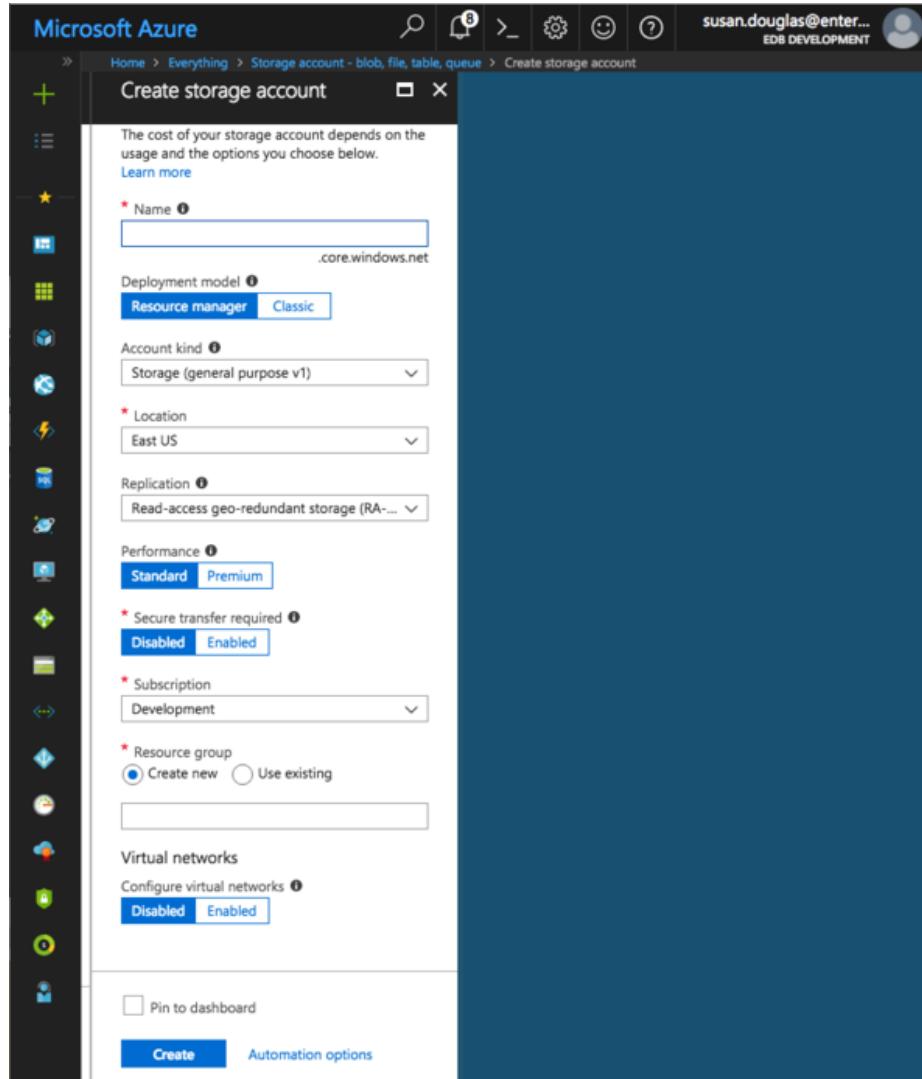
The rule that opens ports 7800 through 7999 provides enough ports for 200 cluster connections; you can extend the upper limit of the port range if more than 200 clusters are required.

#### Creating a Storage Account on Azure

##### **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.



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.

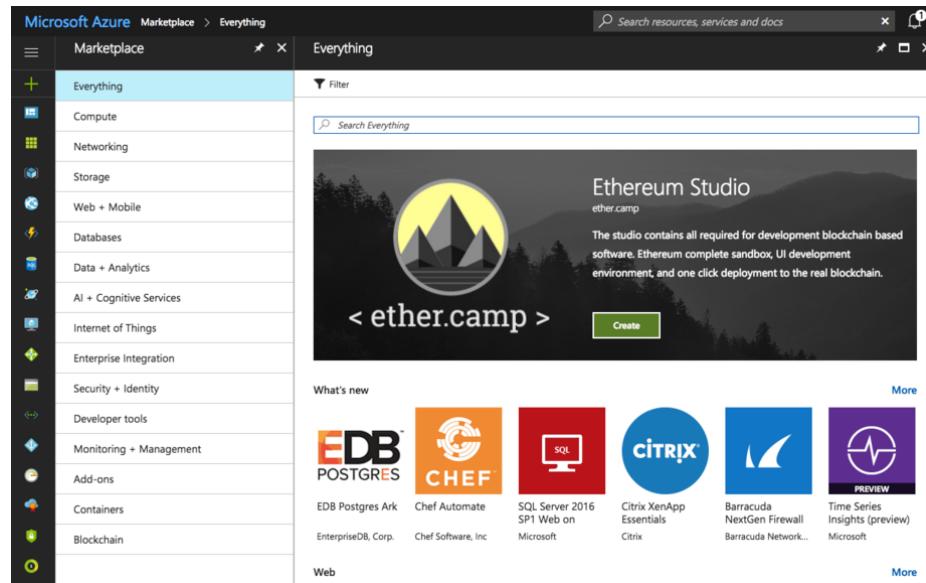
When you've defined your storage account, click **Create**; the Azure dashboard will keep you informed as the storage account is deployed, and send you a notification when the account creation is finished.

For detailed information about defining a storage account, please consult the Azure documentation.

Launching the Ark Console on Azure

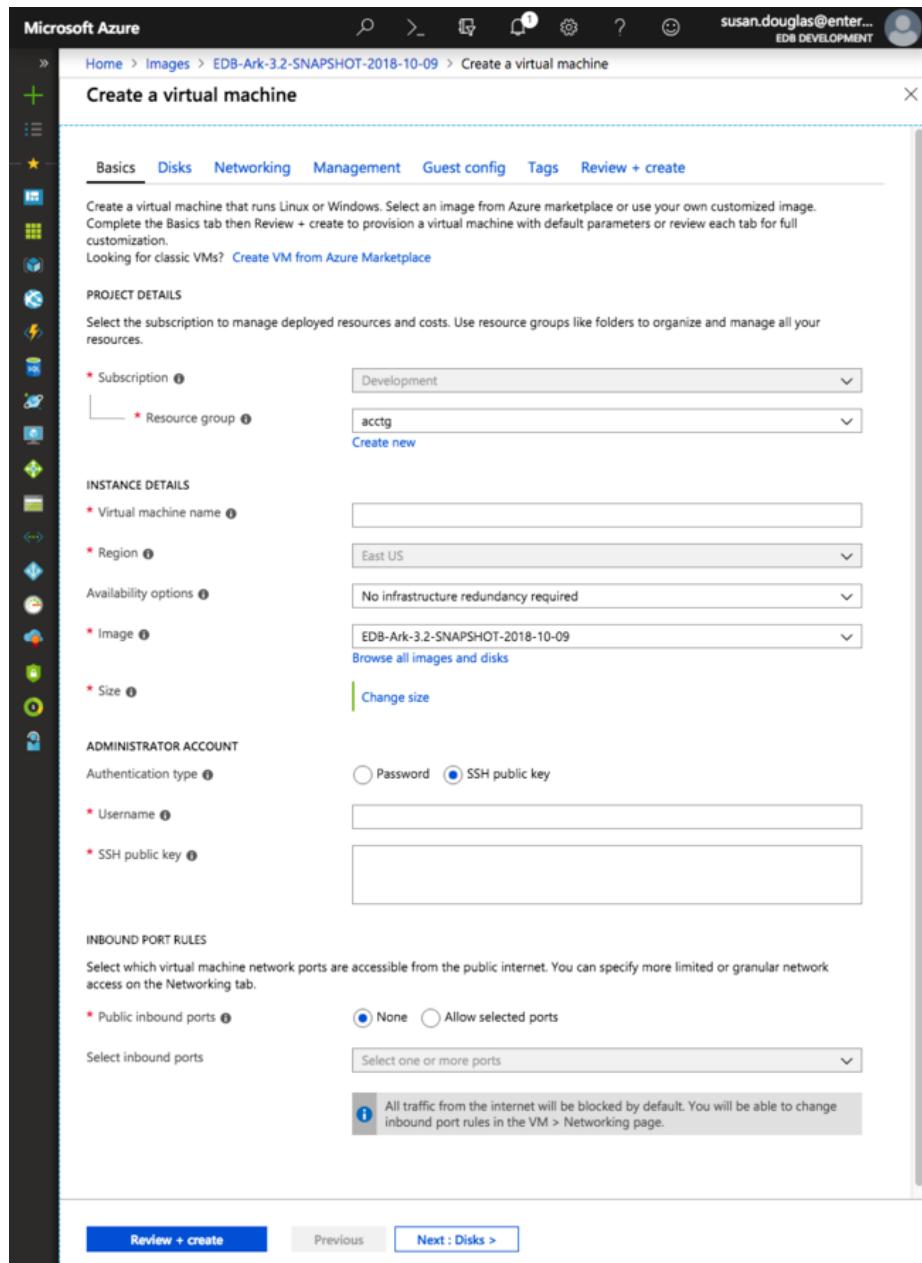
## Launching the Ark Console Instance

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.



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.



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.

The screenshot shows the Microsoft Azure 'Select a VM size' interface. At the top, there's a search bar labeled 'Search by VM size...' and a 'Clear all filters' button. Below that are filter buttons for 'Size : Small', 'Generation : Current', 'Family : General purpose', and 'Premium disk : Supported'. The main area displays 12 of 194 VM sizes, with the following details:

VM SIZE	OFFERING	FAMILY	VCPU	RAM (GB)	DATA DISKS	MAX IOPS	TEMPORARY STOR...
B1ms	Standard	General purpose	1	2	2	800	4 GB
B1s	Standard	General purpose	1	1	2	400	4 GB
B2ms	Standard	General purpose	2	8	4	2400	16 GB
B2s	Standard	General purpose	2	4	4	1600	8 GB
B4ms	Standard	General purpose	4	16	8	3600	32 GB
D2s_v3	Standard	General purpose	2	8	4	3200	16 GB
D4s_v3	Standard	General purpose	4	16	8	6400	32 GB
DS1_v2	Standard	General purpose	1	3.5	4	3200	7 GB
DS2_v2	Standard	General purpose	2	7	8	6400	14 GB
DS2_v2	Promo	General purpose	2	7	8	6400	14 GB
DS3_v2	Standard	General purpose	4	14	16	12800	28 GB
DS3_v2	Promo	General purpose	4	14	16	12800	28 GB

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.

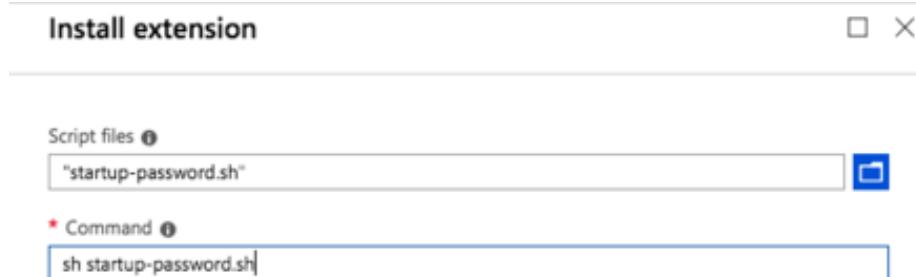
Use fields on the **Guest config** panel to provide an extension that sets the Ark console deployment password. Create a file on your local system named startup-password.sh that contains the following text:

```
|#!/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

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.



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.

The screenshot shows the Microsoft Azure portal interface for creating a new virtual machine. The top navigation bar includes links for Home, Images, EDB-Ark-3.2-SNAPSHOT-2018-10-09, and Create a virtual machine. The user's email, susan.douglas@enter..., and name, EDB DEVELOPMENT, are displayed on the right. The main page title is 'Create a virtual machine'. A green banner at the top indicates 'Validation passed'. Below this, the 'Review + create' tab is selected. The configuration details are listed in sections: **BASICS**, **DISKS**, **NETWORKING**, **MANAGEMENT**, and **GUEST CONFIG**. The **BASICS** section shows the VM name as 'EDB-Ark-3.2-SNAPSHOT-2018-10-09' and the size as 'Standard B1ms' (1 vcpu, 2 GB memory). The **DISKS** section shows 'OS disk type' as Premium SSD and 'Use unmanaged disks' as No. The **NETWORKING** section shows the virtual network as 'apas-cluster-other-resource-grp' and the subnet as 'apas-cluster-other-resource-grpf5e230679'. The **MANAGEMENT** section shows 'Boot diagnostics' as On, 'OS guest diagnostics' as Off, 'Diagnostics storage account' as 'arkapistorageaccount', 'Managed service identity' as Off, and 'Auto-shutdown' as Off. The **GUEST CONFIG** section shows 'Extensions' as 'Custom Script For Linux' and 'Extensions' as No. At the bottom, there are buttons for 'Create' (highlighted in blue), 'Previous', 'Next', and a link to 'Download a template for automation'.

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.

The screenshot shows the Microsoft Azure portal interface. On the left, there's a sidebar with various icons for managing resources. The main content area is titled 'acctg-oct-2018' and shows the 'Overview' tab selected. It provides details such as the resource group (OMG), status (Running), location (East US), and subscription information. Below this, there are two line charts: 'CPU (average)' and 'Network (total)'. The CPU chart shows usage peaking at 4.64% around 11:45 AM. The Network chart shows traffic spikes, with Network In (Avg) at 30.94 MB and Network Out (Sum) at 1.7 MB.

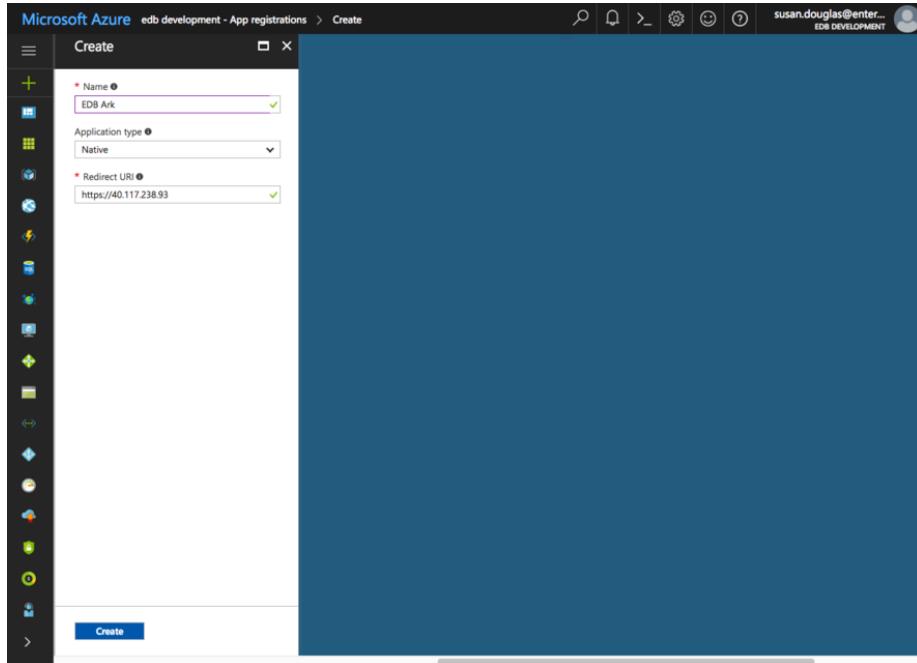
While the deployment finishes, you can register your application in the Azure Active Directory. You will need the Public IP address or DNS name of your server for the registration. To copy the IP address, click the copy icon to the right of the public IP address on the VM details panel.

This screenshot shows the 'App registrations' section within the 'Azure Active Directory' blade of the Microsoft Azure portal. The sidebar on the left has 'App registrations' selected. The main area lists three registered applications: 'EDB Ark', 'acctg-august', and 'acctg-sept'. Each entry includes the application type (Native) and its corresponding Application ID.

DISPLAY NAME	APPLICATION TYPE	APPLICATION ID
EDB Ark	Native	564526f5-d41d-4ba9-993a-e9923861c96c
acctg-august	Native	d591fc74-e0fd-47bf-9bcd-26f692dac7cd
acctg-sept	Native	878236c4-c431-49d2-b830-c46462a0f9a7

After copying the public IP address of your server, select **App registrations** from the Azure Active Directory page. Click the New application registration

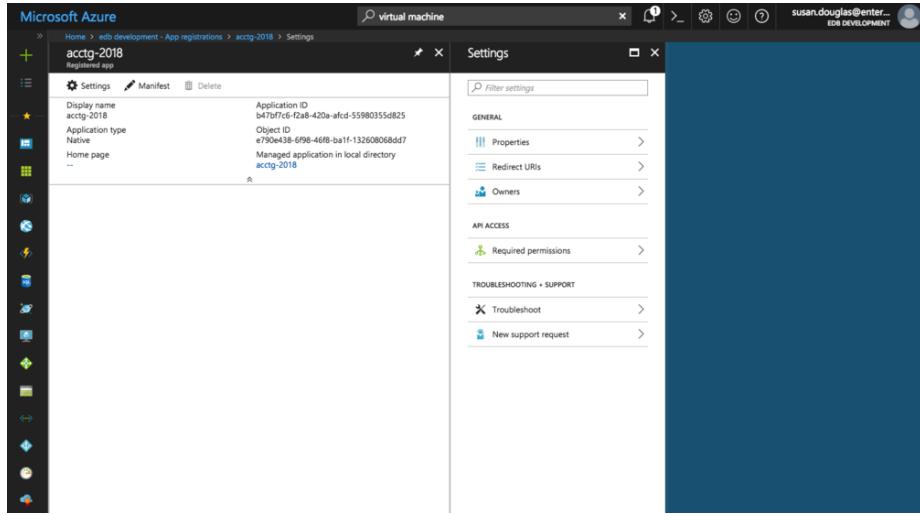
button located on the App registrations detail panel.



Use fields on the **Create** panel to provide information about your application:

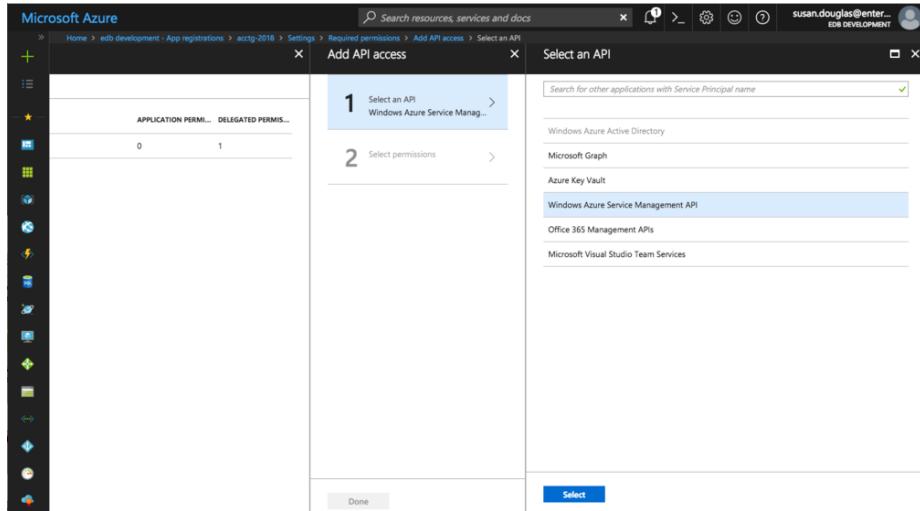
- Provide the application name in the **Name** field.
- Use the drop-down listbox in the **Application type** field to select the Application type; select **Native** for the Ark console application.
- Provide the public IP address of the virtual machine that is hosting the console in the **Redirect URI**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

Click **Create** to register your application.

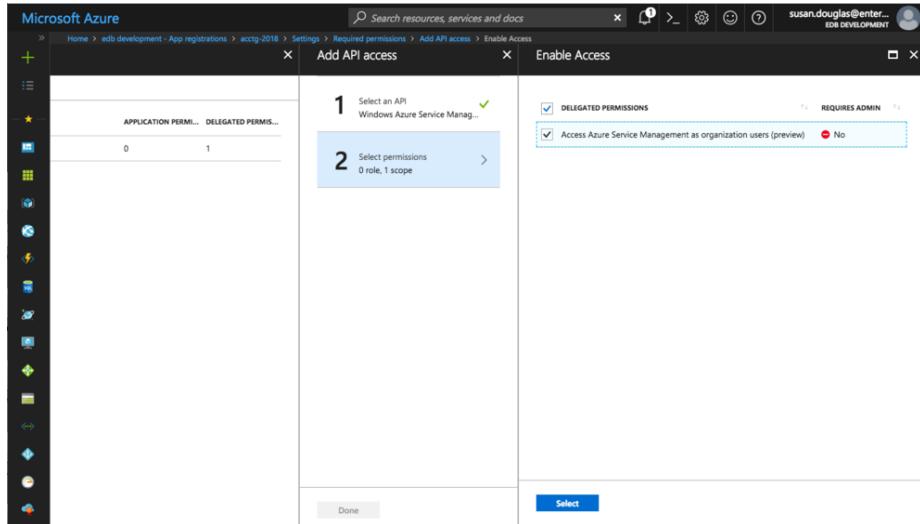


After creating the virtual machine and registering the application, you must adjust the required permissions, allowing the **Windows Azure Service Management API** to connect to your application. This will give the Ark server permission to control Azure services via the Service Management API.

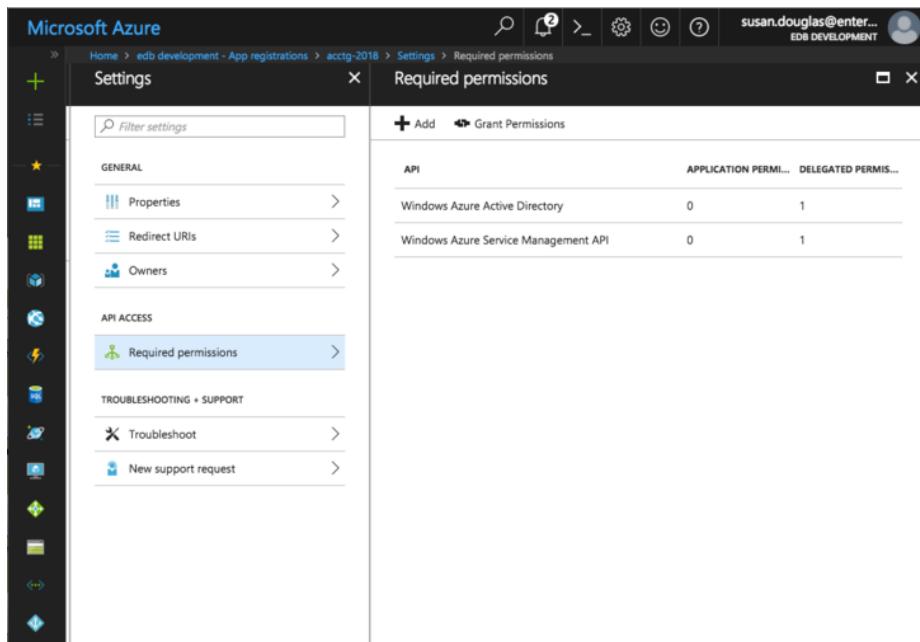
Please note that you must be an Azure Global Administrator to grant permissions required by Ark. Click the **Settings** icon, and then navigate to the **Required permissions** page for the application, and select **+Add**.



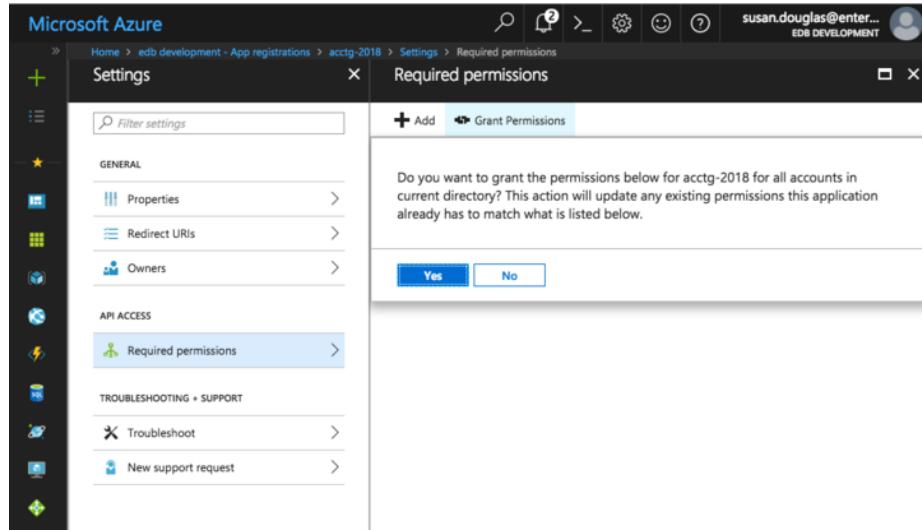
Click **Select an API**, and then highlight Windows Azure Service Management API.



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



Then, click Grant Permissions.



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 scope of Read all users full profiles.

Required permissions		
	+ Add	Grant Permissions
<b>API</b>		
Windows Azure Active Directory	0	1
Windows Azure Service Management API	0	1
Microsoft Graph	0	1

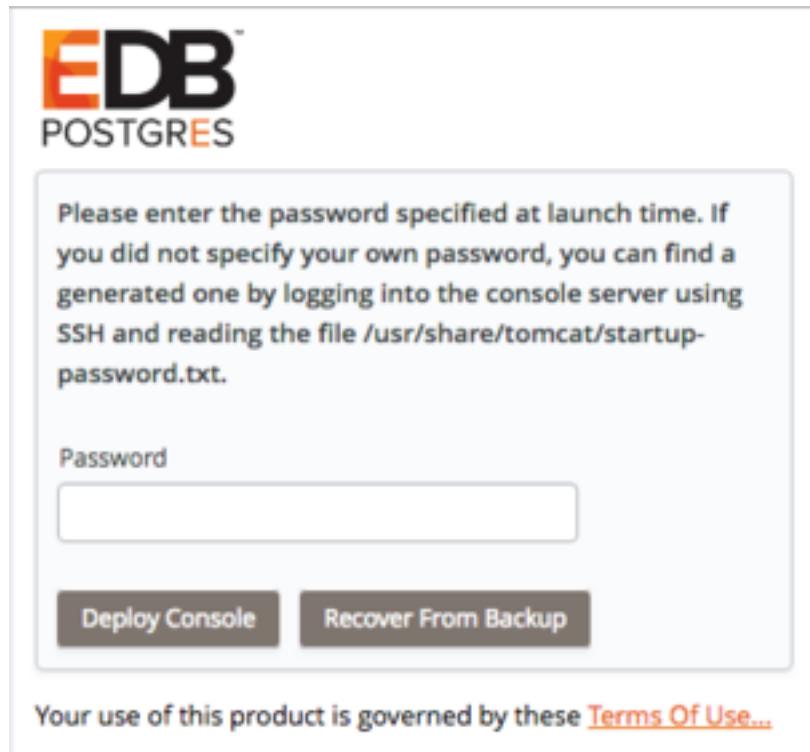
When you're finished granting permissions, the **Required permissions** list should include:

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

Configuring the Ark Console on Azure

## 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.



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.



EDB Ark

Use the following fields to set Ark console properties.

These properties are specific to the Microsoft Azure provider:

Azure Subscription ID	<input type="text"/>
Azure Active Directory ID	<input type="text"/>
Azure Application Registration ID	<input type="text"/>
Service Account ID	<input type="text"/>
Service Account Password	<input type="password"/>
Azure Storage Account	<input type="text"/>

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.

Provide general server properties in the following section:

Console DNS Name	<input type="text"/>
Contact Email Address	<input type="text"/>
Email From Address	<input type="text"/>
Notification Email	<input type="text"/>
Cc From Address	<input type="text"/>
API Timeout	<input type="text"/>
WAL Archive Container	<input type="text"/>
Dashboard Docs URL	<input type="text"/>
Dashboard Hot Topics URL	<input type="text"/>
Enable Console Manager	<input type="text"/>
Enable Postgres Authentication	<input type="text"/>
Template Restrict New Users	<input type="text"/>
Cluster Event Retention Limit	<input type="text"/>

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.

Use the following properties to configure integration with a PEM server:

PEM Server Mode	REMOTE
PEM Server Address	
IP address:	
PEM Server DB Port	
PEM Server API Port	
PEM Server Username	
PEM Server Password	
PEM Sync Mode	DISABLED

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.

## saml Configuring Ark for SAML

Use the following properties to configure the SAML service provider:

SAML Auth Enabled	<input type="text" value="true"/>
SP Entity ID	<input type="text"/>
SP Consumer Service URL	<input type="text"/>
SP Consumer Service Binding	<input type="text"/>
SP Logout Service URL	<input type="text"/>
SP Logout Service Binding	<input type="text"/>
SP Name ID Format	<input type="text"/>
SP Certificate	<input type="text"/>
SP Private Key	<input type="text"/>

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.

IDP Entity ID	<input type="text"/>
IDP Sign On URL	<input type="text"/>
IDP Sign On Service Binding	<input type="text"/>
IDP Logout Service URL	<input type="text"/>
IDP Logout Service Response URL	<input type="text"/>
IDP Single Logout Service Binding	<input type="text"/>
IDP Certificate	<input type="text"/>

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.

Encrypted Name ID	<input type="text"/>
Auth Request Signed	<input type="text"/>
Logout Request	<input type="text"/>
Logout Response Signed	<input type="text"/>
Sign messages	<input type="text"/>
Sign assertions	<input type="text"/>
Sign Metadata	<input type="text"/>
Encrypt Assertions	<input type="text"/>
Encrypt Name ID	<input type="text"/>
Authentication Context	<input type="text"/>
Auth Comparison Parameters	<input type="text"/>
XML validation	<input type="text"/>
Signature Algorithm	<input type="text"/>
Organization Name	<input type="text"/>
Display Name	<input type="text"/>
Organization URL	<input type="text"/>
Organization Language	<input type="text"/>
Technical Contact Name	<input type="text"/>
Technical Email Address	<input type="text"/>
Support Contact Name	<input type="text"/>
SAML Support Email Address	<input type="text"/>

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>

Organization Name	<input type="text"/>
Display Name	<input type="text"/>
Organization URL	<input type="text"/>
Organization Language	<input type="text"/>
Technical Contact Name	<input type="text"/>
Technical Email Address	<input type="text"/>
Support Contact Name	<input type="text"/>
SAML Support Email Address	<input type="text"/>

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.

Use the following properties to enable console backup storage:

Storage Bucket	<input type="text"/>
Console Backup Folder	<input type="text"/>

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.

Use the following properties to change password for DB user:

DB User New Password	<input type="password"/>
DB User Confirm Password	<input type="password"/>

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.

Specify a timezone for the server:

Timezone	<input type="text"/> 
----------	--

Click Save to preserve your edits, validate the properties with the service provider, and configure and deploy the Ark console.

**Save**

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

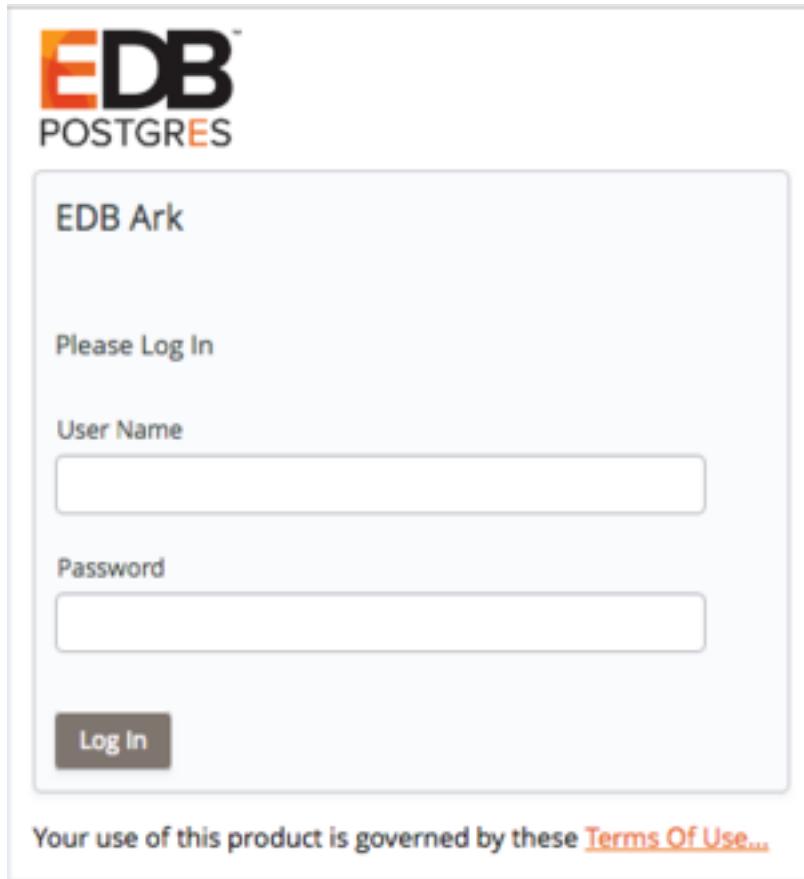
- Use the drop-down listbox in the **Timezone** field to select the timezone that will be displayed by the Ark console.

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 Admin Console on an Azure Host

#### **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.



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.

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 <ark\_using\_admin\_tab>` to create the server .../images and database engines that will be used by non-administrative users.

## 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.



## EDB Ark

Please Log In

Email

Password

[Forgot your password?](#)

[Log In](#)

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

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.

**EnterpriseDB Ark** US EAST 1

Consoles: US EAST 1 Role: ServiceUserRole Logout

**Getting Started**

To begin using EDB Ark, you will want to launch a database cluster.

**Launch DB Cluster**

**Hot Topics**

**EDB Postgres Cloud Database Service Documentation:**

- [EDB User Documentation](#)

EDB Postgres Advanced Server and PostgreSQL version 12 are now available.

- [EDB Postgres Advanced Server Guide v12](#)
- [PostgreSQL Core Documentation v12](#)

**Additional Resources:**

- [Join the EDB Rocks Community!](#)
- [EnterpriseDB Blog](#)
- [EnterpriseDB Product Documentation](#)
- [EnterpriseDB Videos](#)

**EDB Ark V3.5 Tutorials and Documentation**

<a href="#">EDB Ark Getting Started Guide (PDF)</a>	<a href="#">EDB Ark Administrative User Guide (PDF)</a>	<a href="#">EDB Ark API User Guide (PDF)</a>	
<a href="#">EDB Postgres Advanced Server Docs</a>	<a href="#">PostgreSQL Documentation</a>	<a href="#">Database Compatibility for Oracle(R) Developers Guide</a>	<a href="#">Free Training: EDB Ark QuickStart</a>

## 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.

**EnterpriseDB Ark** Role: ServiceUserRole Logout

**Console Manager**

Name for this Console:  API Token for this Console:

**Server Type Administration**

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

Server ID	Server Description	Image ID	Initial User	System Type
C7-ENA	Cent OS 7	ami-d5bf2caa	centos	CentOS

### 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**

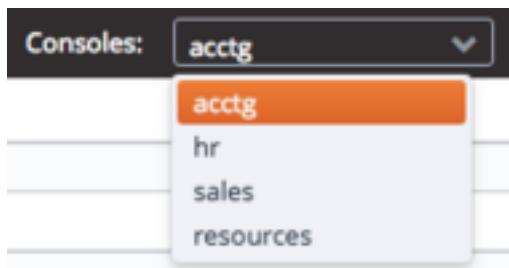
Click the **Edit installation properties** button to open a dialog that allows you to modify the Ark console configuration. For more information, see Editing Installation Properties.

Console Manager

## 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.



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.

A screenshot of the "Console Manager" section of the Admin tab. It includes fields for "Name for this Console" and "API Token for this Console", along with "Save Name" and "Remove Name" buttons. Below these are "Generate API Token" and "Revoke API Token" buttons. A note states "A name for this console is required for these links to be shown." A table header is visible with columns for "Name", "URL", "Region", and "API Token". At the bottom are "Add Console", "Edit Console", and "Delete Console" buttons.

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.

**Console Switcher**

Name for this Console:

**Save Name**   **Remove Name**

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 Console** button; the **Add Console** dialog opens.

**Console Manager**

Name for this Console:

**Save Name**   **Remove Name**

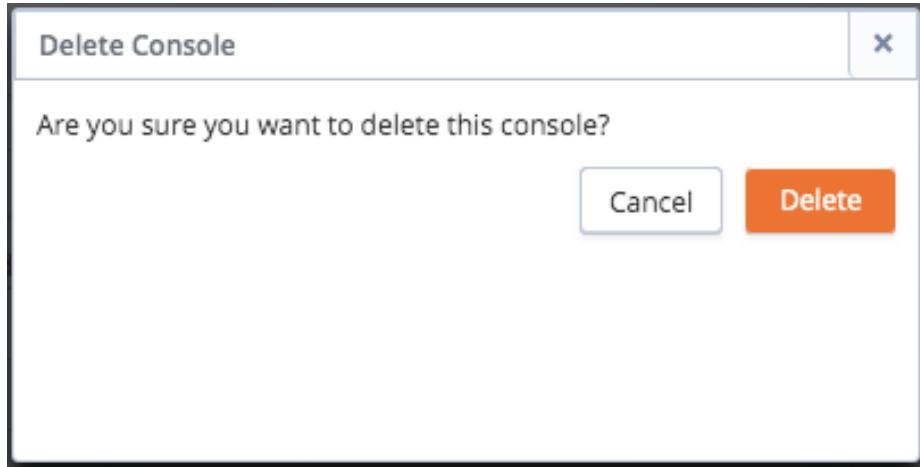
Then, use the **Add Console** dialog to provide information about the console for which you are creating a **Consoles** entry:

- 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 Console** button. After modifying the console details on the **Edit Console** 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 Console** button. A dialog will open, asking you to confirm that you wish to delete the 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.

Federated console Cross-region resources

### 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

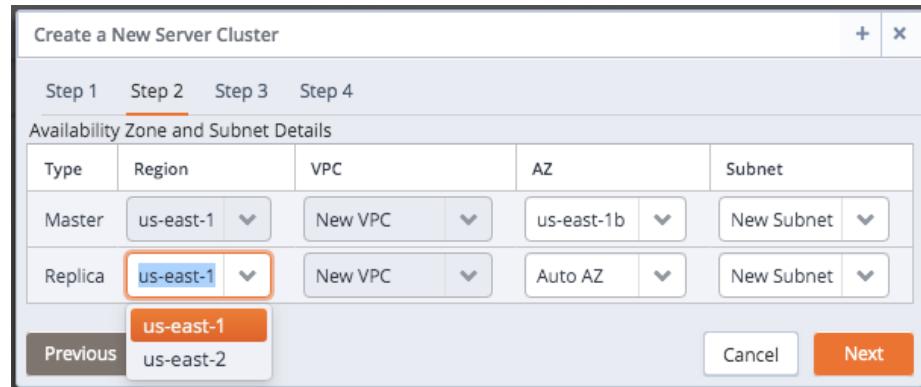
Each federation may only contain one console from a specific region; if you try to add more than one console from a given region to a federation, Ark will return an error.

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.



Type	Region	VPC	AZ	Subnet
Master	us-east-1	New VPC	us-east-1b	New Subnet
Replica	us-east-1	New VPC	Auto AZ	New Subnet

**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:

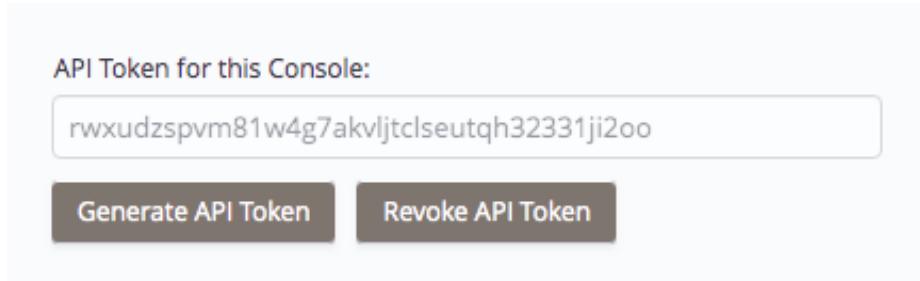
1. Use the **Name for this Console** field to create a name for each console that will be federated. Click within the field and provide a console name; when finished, click the **Save Name** button. The console name will be displayed on the title bar of the Ark console.

**Name for this Console:**

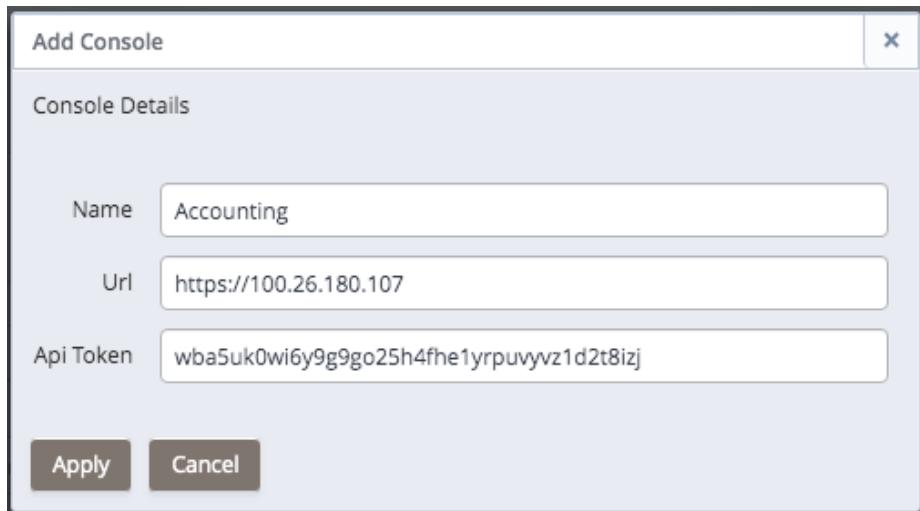
**Resources**

**Save Name**   **Remove 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.



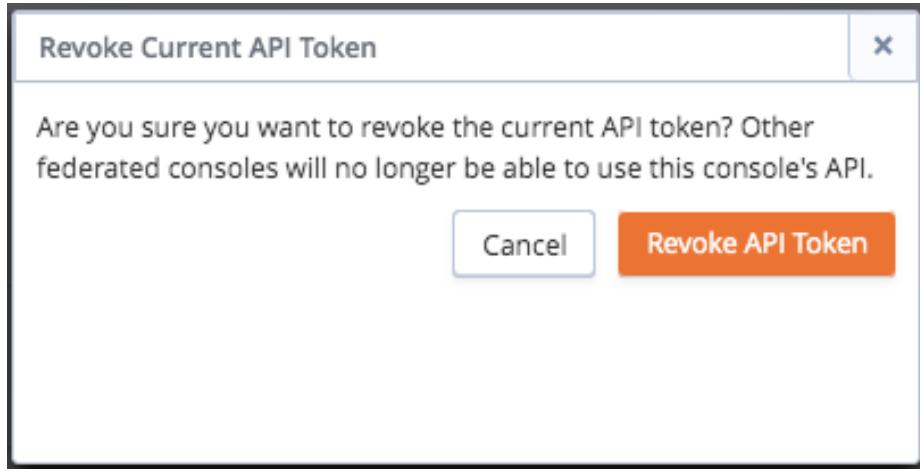
3. Use the Add Console dialog on each console within the federation to register the other consoles with which they will be federated.



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.



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

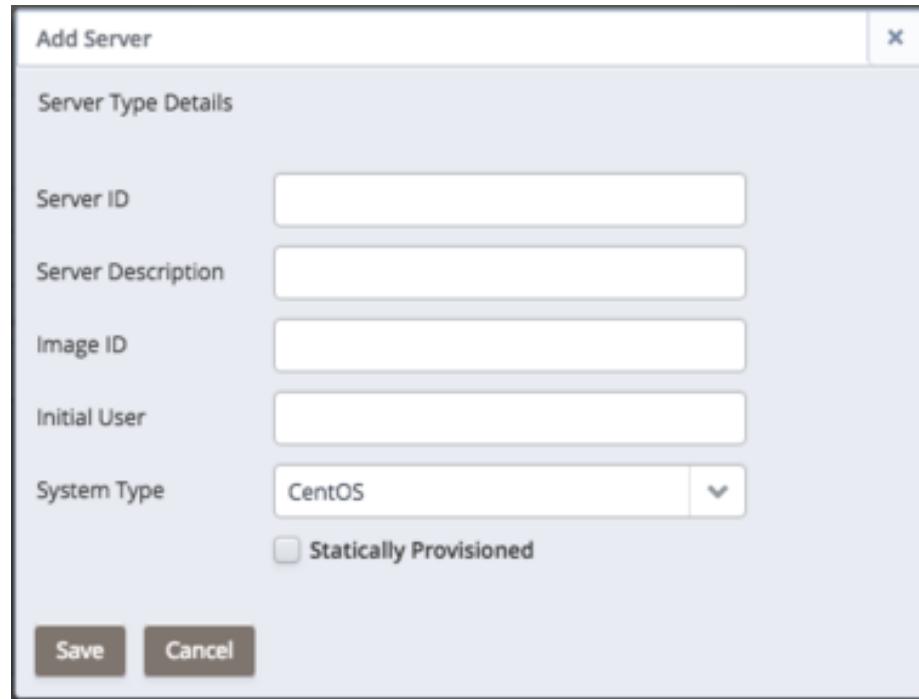
### Managing Server Images

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.

Server Type Administration					
This table allows you to manage base server images which will be provisioned during cluster creation.					
Server ID	Server Description	Image ID	Initial User	System Type	Statically Provisioned
Centos-7	CentOS 7	ami-9887c6e7	centos	CentOS	false
<a href="#">Add Server</a> <a href="#">Edit Server Details</a> <a href="#">Delete Server</a>					

### 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.



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 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`, 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 [Creating a Statically Provisioned Image](#).

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.

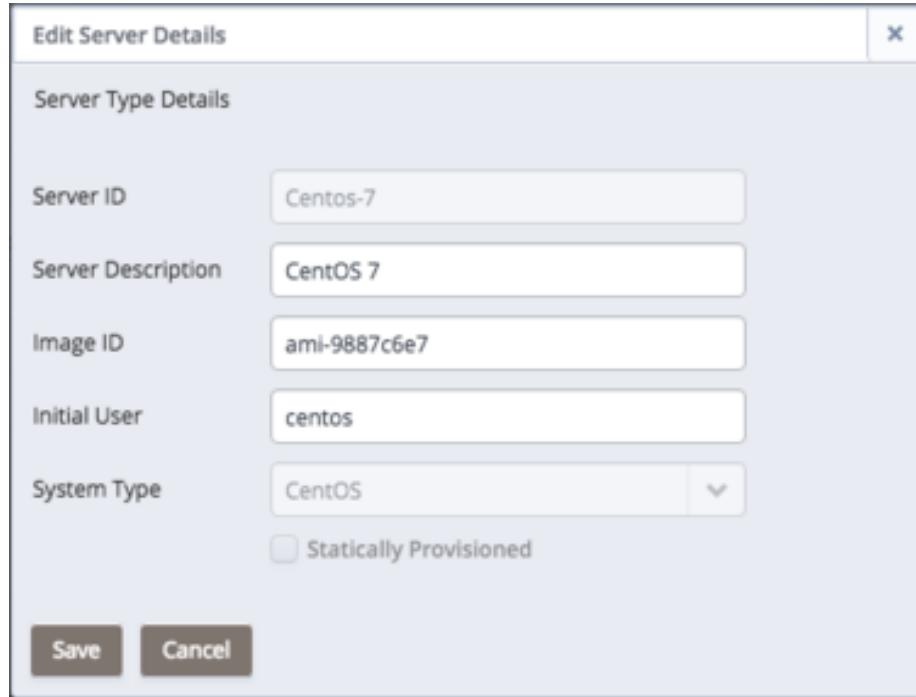
Edit Server Details

Server Type Details

Server ID	Centos-7
Server Description	CentOS 7
Image ID	ami-9887c6e7
Initial User	centos
System Type	CentOS

Statically Provisioned

**Save** **Cancel**



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

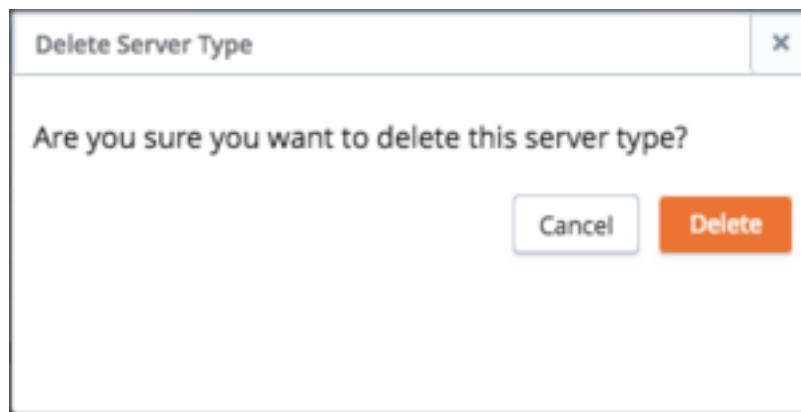
### Deleting a Server

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.

Delete Server Type

Are you sure you want to delete this server type?

**Cancel** **Delete**



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

Error: You can not remove this server type because it is referenced by at least one DB Engine (PPAS\_10\_ARK30)

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

### 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.

ID	Enabled	DB Type	Version	Name	Server Type	RHEL Subscription	Required DB Packages	Opt.
EnterpriseDB PostgreSQL Advanced Server 9.5 64bit on CentOS/RHEL 7								
PPAS_95_CR7_ARK34	false	ppas	9.5	EDB Postgres Advanced Server 9.5 64bit on CentOS/RHEL 7			ppas95-server ppas-pg	
PG_96_CR7_ARK30	false	postgres	9.6	PostgreSQL 9.6 64bit on CentOS / RHEL 7			postgresql96-server pg	
PPAS_96_CR7_ARK34	false	ppas	9.6	EDB Postgres Advanced Server 9.6 64bit on CentOS/RHEL 7			edb-as96-server edb-pg	
PG_10_CR7_ARK30	true	postgres	10	PostgreSQL 10 64bit on CentOS / RHEL 7	C7-ENA		postgresql10-server pg	
PPAS_10_CR7_ARK34	true	ppas	10	EDB Postgres Advanced Server 10 64bit on CentOS/RHEL 7	C7-ENA		edb-as10-server edb-pg	
PG_11_CR7_ARK32	false	postgres	11	PostgreSQL 11 64bit on CentOS / RHEL 7	C7-ENA		postgresql11-server pg	
PPAS_11_CR7_ARK34	false	ppas	11	EDB Postgres Advanced Server 11 64bit on CentOS/RHEL 7	C7-ENA		edb-as11-server edb-pg	
PG_12_CR7_ARK35	false	postgres	12	PostgreSQL 12 64bit on CentOS / RHEL 7			postgresql12-server pg	
PPAS_12_CR7_ARK35	false	ppas	12	EDB Postgres Advanced Server 12 64bit on CentOS/RHEL 7			edb-as12-server edb-pg	

**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://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://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://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://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://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://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

Managing Engine Definitions

#### **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**.

Add Engine

DB Engine Details

ID	<input type="text"/>
DB Type	postgres <input type="button" value="▼"/>
Version	<input type="text"/> <input type="button" value="▼"/>
Name	<input type="text"/>
Server Type	Centos-7 <input type="button" value="▼"/>
RHEL Subscription	<input type="text"/> <input type="button" value="▼"/>
Yum Repo URL(s)	<input type="text"/>
Required DB Packages	<input type="text"/>
Optional Node Packages	<input type="text"/>
<input type="button" value="Save"/> <input type="button" value="Cancel"/>	

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.

**Edit Engine Details**

DB Engine Details

ID	EPAS_12_acctg
DB Type	ppas
Version	12
Name	EPAS 12
Server Type	C7-ENA
RHEL Subscription	
Yum Repo URL(s)	https://USERNAME:PASSWORD@yum.enterprisedb.com/edb-r
Required DB Packages	edb-as12-server edb-pgpool40 edb-as12-pgpool40-extensio
Optional Node Packages	
<input type="checkbox"/> Disabled	
<b>Save</b> <b>Cancel</b>	

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.

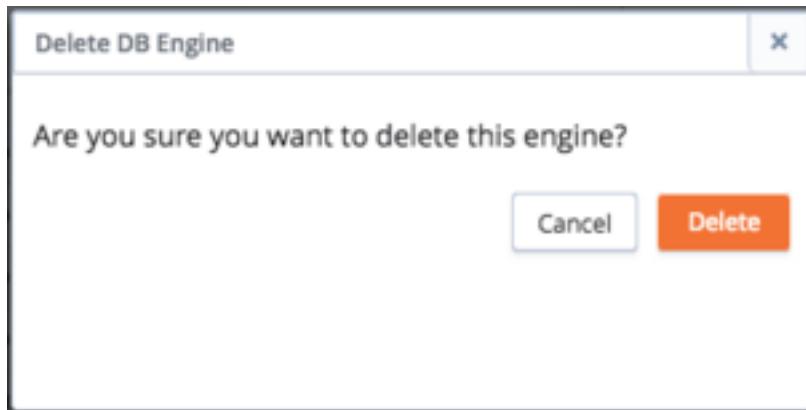
Click the **Save** button to make any changes to the **Edit Engine Details** dialog persistent, or select **Cancel** to exit without modifying the engine definition.

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.



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 Components to a DB Engine**

#### **Adding Supporting Components to a Database Engine Definition**

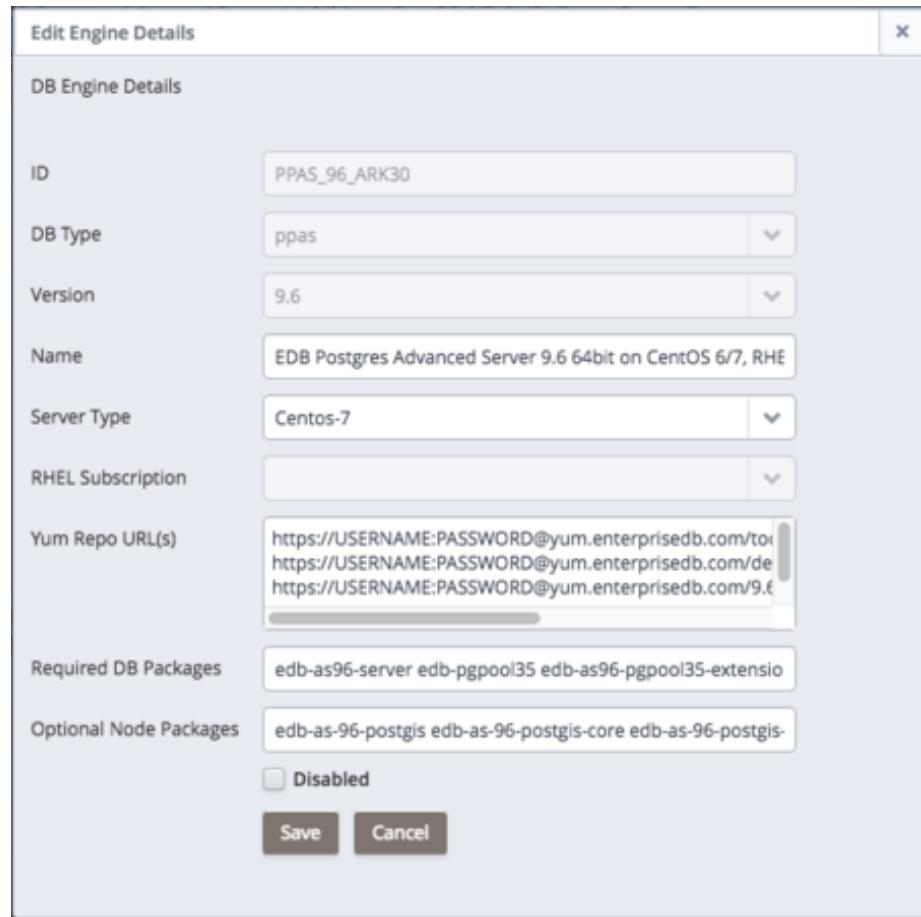
When you create a cluster, you specify the engine that EDB Ark will use when provisioning that cluster. If you modify the engine description, adding the list of RPM packages that will be installed when that engine is provisioned, each node of any cluster provisioned with that engine will include the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



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

### 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.

Template Administration		
This table lets you create, edit, and delete templates.		
Enabled	Template Name	Template Available To Tenants
true	clerk	ark-qmng,ark-dev
true	sales	ark-qmng,ark-dev

Add Template    Edit Template    Delete Template    Refresh

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.

Add Template

Template Details

Template Name

Description

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

Server Class t2.micro

Use Private IP addresses

VPC New VPC

Number Of Nodes 1

Storage GB 1

Encrypted

EBS Optimized

IOPS 0

Perform OS and software update

Number of backups 1

Backup Window (GMT-05:00) 12:00am - 2:00am

Continuous Archiving (Point-in-Time Recovery)

Select Scaling Options

Manually Scale Replicas

Manually Scale Storage

Auto Scale Replicas

Auto Scale Storage

Select Roles

edb-ark-hans

Disabled

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 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.



Red Hat Subscription Management

### 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 subscription 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.

RHEL Subscription Management												
This table allows you to manage RHEL subscriptions												
Subscription ID	User Name	Server URL	Base URL	Org	Environment	Name	Auto Attach	Activation Key	Service Level	Release	Force	Type
Admin	carol.smith@enterprisedb.com	subscriptions.rhn.redhat.com	https://cdn.redhat.com	EnterpriseDB	Accounting	acctg	true	7488hg955	Standard	RHEL7	false	system
Sales	bob.king@enterprisedb.com	subscriptions.rhn.redhat.com	https://cdn.redhat.com	EnterpriseDB	sales	sales	true	9945hko223	Standard		false	system

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

Add RHEL Subscription

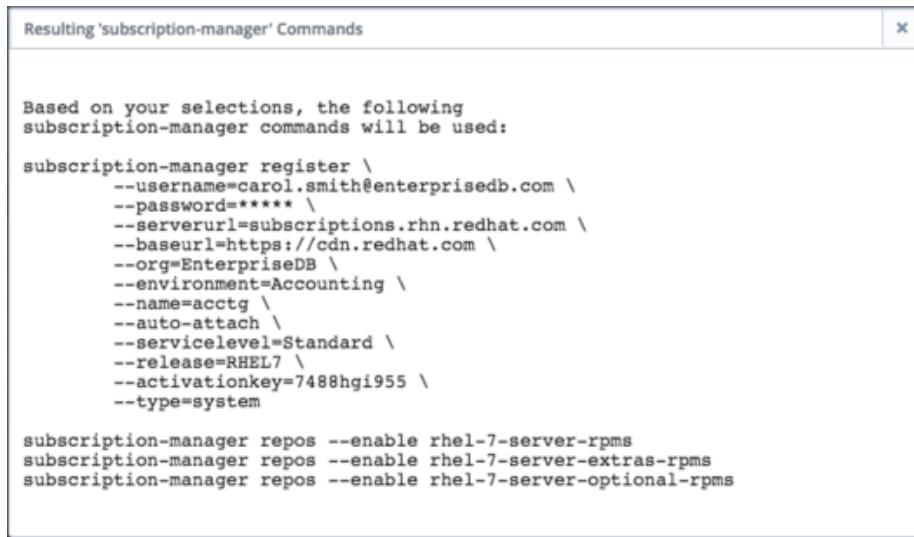
RHEL Subscription Details

Subscription ID	
Username	
Password	
Server URL	
Base URL	
Org	
Environment	
Name	
Activation Key	
<input checked="" type="checkbox"/> Auto-attach	
Pool	
<input type="checkbox"/> Auto	
Quantity	0
Service Level	
Release	
<input type="checkbox"/> Force	
Type	
Required Repos	rhel-7-server-rpms rhel-7-server-extras-rpm rhel-7-server-optional-rpms
Additional Repos	
Disabled Repos	
<input type="button" value="Save"/> <input type="button" value="Cancel"/>	

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.



The screenshot shows a window titled "Resulting 'subscription-manager' Commands". Inside the window, there is a message: "Based on your selections, the following subscription-manager commands will be used:". Below this message is a block of command-line text:

```
subscription-manager register \
--username=carol.smith@enterprisedb.com \
--password=***** \
--serverurl=subscriptions.rhn.redhat.com \
--baseurl=https://cdn.redhat.com \
--org=EnterpriseDB \
--environment=Accounting \
--name=acctg \
--auto-attach \
--servicelevel=Standard \
--release=RHEL7 \
--activationkey=7488hg1955 \
--type=system

subscription-manager repos --enable rhel-7-server-rpms
subscription-manager repos --enable rhel-7-server-extras-rpms
subscription-manager repos --enable rhel-7-server-optional-rpms
```

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.

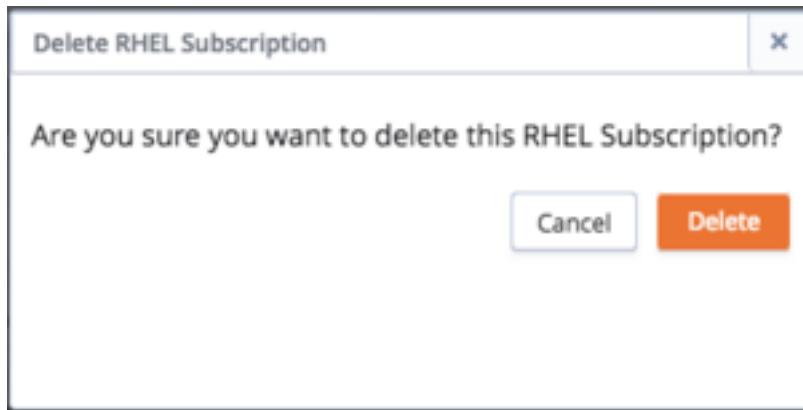
After modifying the subscription definition, click **Save** to preserve your changes and exit the dialog; to exit without saving, click the **Cancel** button. Please note that changes made to a definition are applied only to those instances that are created after the changes are saved; changes are not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



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

### 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.



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.

Role ARN: arn:aws:iam::325007900792:role/susan

Role description: Allows EC2 instances to call AWS services on your behalf.

Instance Profile ARNs: arn:aws:iam::325753300792:instance-profile/susan

Path: /

Creation time: 2017-05-15 11:35 EST

Give this link to users who can switch roles in the console: <https://signin.aws.amazon.com/switchrole?roleName=susan&account=cloud>

**Permissions** **Trust relationships** **Access Advisor** **Revoke sessions**

You can view the trusted entities that can assume the role and the access conditions for the role. Show policy document

**Edit trust relationship**

**Trusted entities**  
The following trusted entities can assume this role.  
The identity provider(s) ec2.amazonaws.com  
The account 325753300792

**Conditions**  
The following conditions define how and when trusted entities can assume the role.

Condition	Key	Value
StringEquals	sts:ExternalId	3eb26a74-00ae-4baf-8dfc-f52227bbf8b

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

Add Role

Enter the complete Role ARN and External ID for the role.

Role Arn:

External Id:

Apply Cancel

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.

The screenshot shows the 'User Administration' section of a software interface. At the top, there is a table with columns: ID, First Name, Last Name, Admin, Enabled, Templates Only, Role, External ID, Clusters, Snapshots, and Last Login. The table contains five rows of user data. Below the table is a large empty space. At the bottom of the page, there is a footer bar with several buttons: Delete Clusters, Delete Snapshots, Add User, Edit User, Delete User, Refresh, Show Logged In Users, Wall Message, Display Message, and Remove Message.

ID	First Name	Last Name	Admin	Enabled	Templates Only	Role	External ID	Clusters	Snapshots	Last Login
alan.james@enterprisedb.com	Alan	James	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	edb-ark-hans	515bd1ec-ebb9-4579-9961-edc110b3dd55	0	0	
susan.douglas@enterprisedb.com	Susan	Douglas	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	edb-ark-hans	515bd1ec-ebb9-4579-9961-edc110b3dd55	0	0	Dec 11, 2018 08:53
carol.smith@enterprisedb.com	Carol	Smith	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	edb-ark-hans	515bd1ec-ebb9-4579-9961-edc110b3dd55	0	0	
bob.king@enterprisedb.com	Bob	King	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	edb-ark-hans	515bd1ec-ebb9-4579-9961-edc110b3dd55	0	0	
scott.ward@enterprisedb.com	Scott	Ward	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	edb-ark-hans	515bd1ec-ebb9-4579-9961-edc110b3dd55	0	0	

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.

**Add User**

**User Details**

Login	<input type="text"/>
First Name	<input type="text"/>
Last Name	<input type="text"/>
<input type="checkbox"/> Admin	
<input checked="" type="checkbox"/> Enabled	
<input type="checkbox"/> Templates Only	
Password	<input type="password"/>
Verify Password	<input type="password"/>
Role	<input type="text"/> <input type="button" value="▼"/>

**Save** **Cancel**

The screenshot shows a modal dialog titled "Add User". The title bar has a close button ("X"). The main area is titled "User Details". It contains fields for "Login" (empty), "First Name" (empty), "Last Name" (empty), and three checkboxes: "Admin" (unchecked), "Enabled" (checked), and "Templates Only" (unchecked). Below these are fields for "Password" and "Verify Password" (both empty). A "Role" dropdown menu is open, showing a single option. At the bottom are "Save" and "Cancel" buttons.

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 dropdown 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 AWS Host

### **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.

User Administration										
ID	First Name	Last Name	Admin	Enabled	Templates Only	Role	External ID	Clusters	Snapshots	Last Login
alan.james@enterprisedb.com	Alan	James	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	edb-ark-hans	515bd1ec-ebb9-4579-9961-edc110b3dd55	0	0	
susan.douglas@enterprisedb.com	Susan	Douglas	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	edb-ark-hans	515bd1ec-ebb9-4579-9961-edc110b3dd55	0	0	Dec 11, 2018 08:53
carol.smith@enterprisedb.com	Carol	Smith	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	edb-ark-hans	515bd1ec-ebb9-4579-9961-edc110b3dd55	0	0	
bob.king@enterprisedb.com	Bob	King	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	edb-ark-hans	515bd1ec-ebb9-4579-9961-edc110b3dd55	0	0	
scott.ward@enterprisedb.com	Scott	Ward	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	edb-ark-hans	515bd1ec-ebb9-4579-9961-edc110b3dd55	0	0	

Buttons at the bottom:

- Delete Clusters
- Delete Snapshots
- Add User
- Edit User
- Delete User
- Refresh
- Show Logged In Users

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

##### 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.

ID	First Name	Last Name	Admin	Enabled	Templates Only	Clusters	Snapshots	Last Login
Ark.service@enterprisedb.com			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	0	0	Dec 11, 2018 08:28

**Edit User**   **Refresh**  
**Show Logged In Users**

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.

#### Display Connected Users

## Displaying Connected Users

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



The dialog displays:

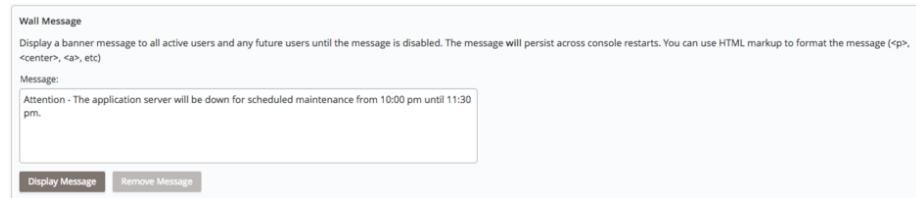
- The current number of empty sessions; an empty session is an http session with the server that is not associated with a logged-in user.
- The current number of sessions with a logged-in user.
- A list of the currently logged-in users.

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

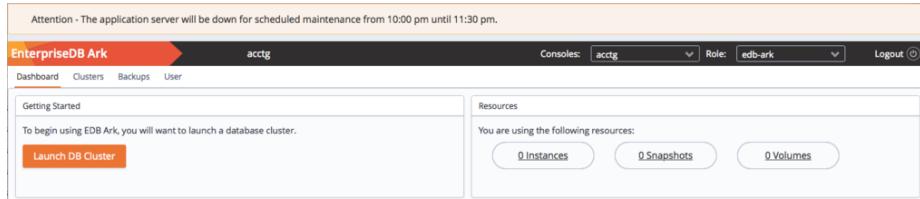
## Wall Message

### 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.



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.

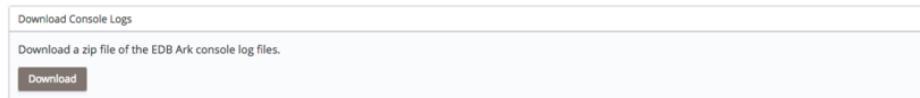


Use the **Remove Message** button to remove the wall banner.

#### Console Logs

##### 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.



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.

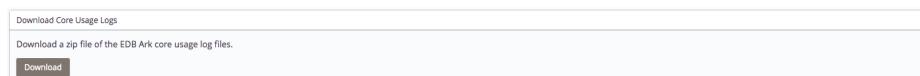
You can include the `-F` option to instruct the `tail` utility to display only the last 10 lines of the log file, and new log file entries as they are added to the file:

```
tail -F file_name
```

#### Core Usage Logs

##### 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.



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.

Manual Backup of the Console

### Taking a Manual Backup of the Console

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

The screenshot shows a simple interface for requesting a manual backup. At the top, it says "Backup Ark Console". Below that is a message: "Manually request a console backup. It may take a few moments for the backup to complete.". At the bottom is a prominent blue button labeled "Backup Now".

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

### Editing Installation Properties

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

The screenshot shows a panel titled "Edit Installation Properties". It contains a message: "This action will bring up a form to allow changes to the Ark installation properties. A server restart is not required to make these changes." At the bottom is a blue button labeled "Edit Installation Properties".

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.

---

## 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.

EnterpriseDB Ark

acctg

Consoles: acctg Role: edb-ark Logout

Choose table/view dbengine

id	engine_id	eol	name	optional_pkgs	required_pkgs
11	PPAS_96_ARK30	false	EDB Postgres Advanced Server 9.6 64bit on CentOS 6/7, RHEL 7	edb-as-96-postgis edb-as-96-postgis-core edb-as-96-postgis-docs edb-as-96-postgis-utils	edb-as96-ser
2	PG_93_C7_ARK30	true	PostgreSQL 9.3 64bit on CentOS / RHEL 7		postgres93
3	PG_94_C6_ARK30	true	PostgreSQL 9.4 64bit on CentOS		postgres94
4	PG_94_C7_ARK30	true	PostgreSQL 9.4 64bit on CentOS / RHEL 7		postgres94
5	PPAS_94_ARK30	true	EDB Postgres Advanced Server 9.4 64bit on CentOS 6/7, RHEL 7		ppas94-serv
6	PG_95_C6_ARK30	true	PostgreSQL 9.5 64bit on CentOS 6		postgres95
7	PG_95_C7_ARK30	true	PostgreSQL 9.5 64bit on CentOS / RHEL 7		postgres95
8	PPAS_95_ARK30	true	EDB Postgres Advanced Server 9.5 64bit on CentOS 6/7, RHEL 7		ppas95-serv
1	PG_93_C6_ARK30	true	PostgreSQL 9.3 64bit on CentOS 6		postgres93
10	PG_96_C7_ARK30	true	PostgreSQL 9.6 64bit on CentOS / RHEL 7		postgres96
12	PG_10_C6_ARK30	true	PostgreSQL 10 64bit on CentOS		postgres10
13	PG_10_C7_ARK30	true	PostgreSQL 10 64bit on CentOS / RHEL 7		postgres10
14	PPAS_10_ARK30	true	EDB Postgres Advanced Server 10 64bit on CentOS 6/7, RHEL 7		edb-as10-ser

Contact information for enabled users  
Contact information for all users

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

Choose table/view

activation

- activation
- attachedvolume
- backups
- consoleurl
- dbengine
- instances
- nodestatistics
- pcshistory
- property
- rhelrepo
- rhelsubscription
- serverimage
- snapshots
- template

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

### 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.

[Contact information for enabled users](#)

[Contact information for all users](#)

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).

`ark_dba_tables`

---

## 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.



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.css
```

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

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

### **The images Directory**

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

```
/usr/share/tomcat/webapps/PPCDConsole/VAADIN/themes/pcsconsole/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/loginscre
```

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

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

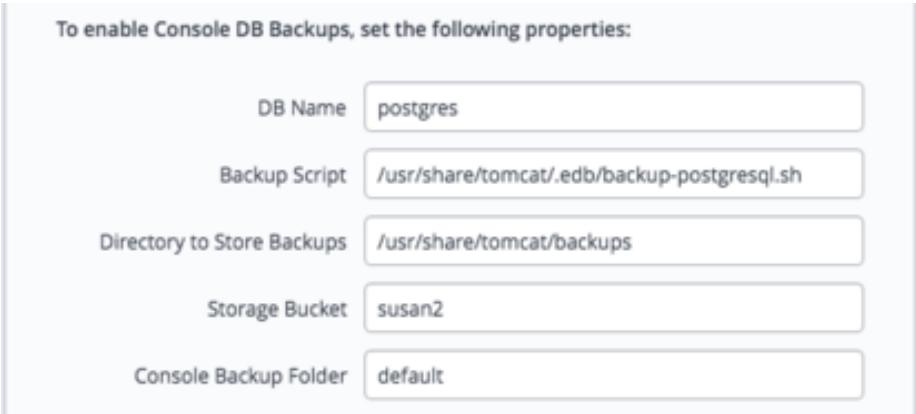
If the system hosting the application database fails, then all information about the console database and registered users will be lost unless you have retained a backup.

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.

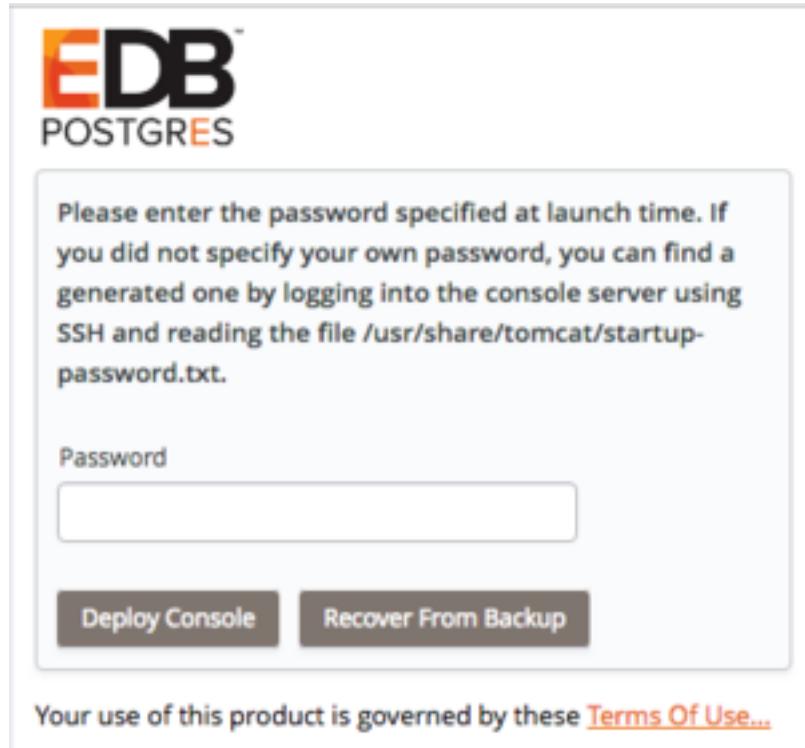
The backup directory specified should not reside on the console VM's root disk; your backup would be lost in the event of a VM failure. You should consider mounting an external volume to the console VM, and writing console database backups to that volume.

- 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.



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\\;; c
```

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.

---

## 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.

#### Security Group, Modifying

#### **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.

Type	Protocol	Port Range	Source
PostgreSQL	TCP	5432	Custom 54.159.105.84/32
Custom TCP Rule	TCP	9999	Custom 0.0.0.0/0
Custom TCP Rule	TCP	7800 - 7802	Custom 54.159.105.84/32
SSH	TCP	22	Custom CIDR, IP or Security Gr

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.

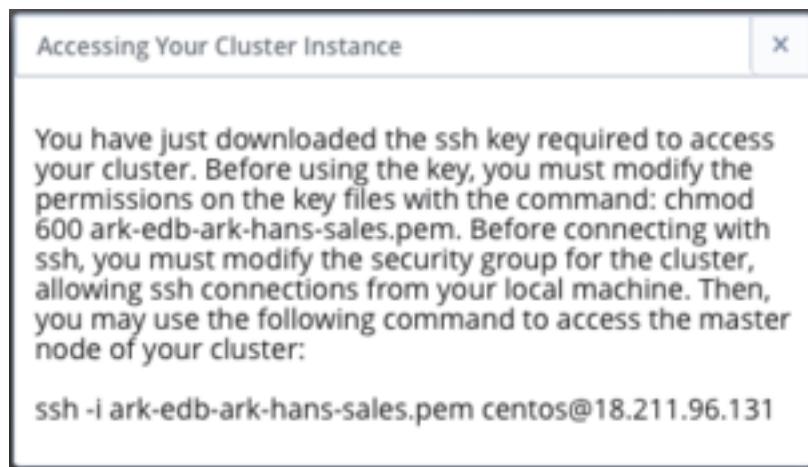
Please consult the Amazon documentation for detailed information about managing the security group for a virtual private cloud.

## SSH Connections

### 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.



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
```

iptables Rules

### 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.

---

## 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
<code>-n</code>	The database server type
<code>-v</code>	The database server version
<code>-u</code>	If true, Ark will invoke the yum update command and update the currently installed software packages.
<code>-c</code>	When set to true, the script will configure and enable required RHEL repositories.
<code>-r</code>	The repository address (and if applicable, credentials) for provisioning. Include the <code>-r</code> flag once for each repository.
<code>-p</code>	A list of the packages that will be installed in the image.
<code>-o</code>	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-pgpool135 edb-as10-pgpool135-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.

---

## 13 EDB Postgres 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 detected with the backup process.
Database State Changed to <i>db_state</i>	The MASTER /REPLICA database service state has changed to <i>db_state</i> .
Load Balancer Port Error	The MASTER /REPLICA database service port is not responding.
Load Balancer Port Notification	The MASTER /REPLICA database service port has been notified.
Continuous Archiving State Changed to <i>db_state</i>	Continuous Archiving on the master/relica node has changed to <i>db_state</i> .
Continuous Archiving State Changed to <i>db_state</i>	A problem was detected with continuous archiving.
Data Storage Scaling <i>cluster_name</i>	Data storage is being added to cluster <i>cluster_name</i> .
Data storage scaling for cluster <i>cluster_name</i> has been suspended	Data storage scaling for cluster <i>cluster_name</i> has been suspended.
Rebuild of primary node in cluster <i>cluster_name</i>	The primary server, node id <i>instance_id</i> is currently being rebuilt.

Subject	Body
Replacement of primary node in cluster <i>cluster_name</i>	The primary server, node id <i>instance_id</i>
Rebuild of replica node in cluster <i>cluster_name</i>	The replica server, node id <i>instance_id</i> i
Replica promotion failed in cluster <i>cluster_name</i>	Replica promotion failed. Performing reb
Replica promotion failed in cluster <i>cluster_name</i>	Replica promotion failed. Node id: <i>insta</i>
WARNING: Connectivity Issue with instance <i>region/instance_id</i>	WARNING: The EDB Ark cluster mana
(PITR) Base Backup of cluster <i>cluster_name</i> failed	The automatic
Backup of cluster <i>cluster_name</i> failed	The automatic
WAL Archive Storage Container Created	A storage container (bucket) named <i>buck</i>
Termination of cluster <i>cluster_name</i> completed.	The termination of cluster <i>cluster_name</i>
WARNING: Termination Protection <i>instance_id</i> .	The system was not able to terminate in
OS/SW update PASSED on node <i>instance_id</i> .	Yum update results for node: <i>dns_name</i>
OS/SW update FAILED on node <i>instance_id</i> .	Yum update results for node: <i>dns_name</i>
OS/SW Status is now: <i>status</i>	The OS/SW status on node <i>dns_name</i> c
OS/SW Status is now: <i>status</i>	The OS/SW status on node <i>dns_name</i> c
Unable to delete Security Group <i>group_name</i>	The system was not able to delete the Se
Volume attachment failed in cluster <i>cluster_name</i>	The message body contains error text di
Ark Synchronization Error With PEM Server	The Ark console has encountered an erro
Reboot of cluster <i>cluster_name</i> in progress	OS/SW update completed successfully, r

---

## 14 EDB Postgres 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: mailto:support@enterprisedb.com
- or by phone: +1-732-331-1320 or 1-800-235-5891 (US Only)

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

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

You are always welcome to log an issue via email; when time permits, our customer support experts will respond to inquiries from customers that have not purchased support. Postgres documentation and helpful tutorials is always available from the **Dashboard** tab of the Ark console.

---

## 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:ModifyInstanceStateAttribute",  
                "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"
}
]

```

}

---

## 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"  
                }  
            }  
        }  
    ]  
}
```

---

## 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": [
    "*"
  ]
}
]
}

```

---

## 18 Conclusion

EDB Postgres Ark Administrator's Guide

Copyright © 2007 - 2019 EnterpriseDB Corporation.

All rights reserved.

EnterpriseDB Corporation

34 Crosby Drive, Suite 201, Bedford, MA 01730, USA

T +1 781 357 3390 F +1 978 467 1307 E

[info@enterprisedb.com](mailto:info@enterprisedb.com)

[www.enterprisedb.com](http://www.enterprisedb.com)

- EDB designs, establishes coding best practices, reviews, and verifies input validation for the logon UI for EDB Postgres Ark where present. EDB follows the same approach for additional input components, however the nature of the product may require that it accepts freeform SQL, WMI or other strings to be entered and submitted by trusted users for which limited validation is possible. In such cases it is not possible to prevent users from entering incorrect or otherwise dangerous inputs.
- EDB reserves the right to add features to products that accept freeform SQL, WMI or other potentially dangerous inputs from authenticated, trusted users in the future, but will ensure all such features are designed and tested to ensure they provide the minimum possible risk, and where possible, require superuser or equivalent privileges.
- EDB does not warrant that we can or will anticipate all potential threats and therefore our process cannot fully guarantee that all potential vulnerabilities have been addressed or considered.