AWS Cloud Migration

Target VPC for This Migration

Plaintext

```
+---------------------------------------------------+
|           TARGET VPC (10.0.0.0/16)                |
|---------------------------------------------------|
|  AZ-A (AZ 1)                 | AZ-B (AZ 2)        |
|---------------------------------------------------|
|  [Public Subnet 0]           | [Public Subnet 1]  |
|  (EIP 0 Assoc.)              | (EIP 1 Assoc.)     |
|       |                      |       |            |
|       v                      |       v            |
|  [NAT Gateway 0] ------------+ [NAT Gateway 1]    |
|  (Outbound ONLY)             | (Outbound ONLY)    |
|       |                      |       |            |
|       v                      |       v            |
|  [Private Route Table 0] (RT)| [Private Route Table 1] (RT) |
|  (Routes to NAT G/W 0)       | (Routes to NAT G/W 1)        |
|---------------------------------------------------|
|  | (RT Association)          | | (RT Association) |
|  v                           | v                  |
|  [Pvt Web Subnet 00]         | [Pvt Web Subnet 10] |
|  (App Servers)               | (App Servers)      |
|                              |                    |
|  [Pvt DB Subnet 01]          | [Pvt DB Subnet 11]  |
|  (DB Servers)                | (DB Servers)       |
+---------------------------------------------------+
```

Target Environment (TargetVPCName)
The target environment is a multi-AZ VPC configured as the destination for the migration.

- AWS::EC2::VPC (TargetVPC): A VPC with the CIDR 10.0.0.0/16 for the migration target environment.
- AWS::EC2::Subnet (PublicSubnet0, PublicSubnet1): Two public subnets in different AZs for internet-facing resources.
- AWS::EC2::Subnet (PrivateSubnet00, PrivateSubnet10): Two private subnets in different AZs for web/application servers.
- AWS::EC2::Subnet (PrivateSubnet01, PrivateSubnet11): Two private subnets in different AZs for database servers.
- AWS::EC2::IGW: A gateway to allow traffic between the Target VPC public subnets and the public internet.
- AWS::EC2::EIP (ElasticIP0, ElasticIP1): Two Elastic IPs to be associated with the NAT Gateways for static public IP addresses.
- AWS::EC2::NatGateway (NATGateway0, NATGateway1): Two NAT Gateways in the public subnets, enabling instances in the private subnets to initiate outbound internet connections.
- AWS::EC2::RouteTable (PublicRouteTable): A route table for the public subnets, directing internet traffic through the InternetGateway.
- AWS::EC2::RouteTable (PrivateRouteTable0, PrivateRouteTable1): Two separate route tables for the private subnets, one for each AZ, directing internet traffic through its respective NAT Gateway.
- AWS::EC2::NetworkAcl (PublicNetworkAcl): A network access control list applied to the public subnets to control traffic flow.

APPLICATION DISCOVERY PHASE (ADS)

*** as of Nov.7th ADS is deprecated and replaced by AWS Transform*

Agent will be installed and run on all the on-premise nodes to get the report as to:

what is running on the servers
what's the recommended ec2 instance needed
what are dependencies on these etc.

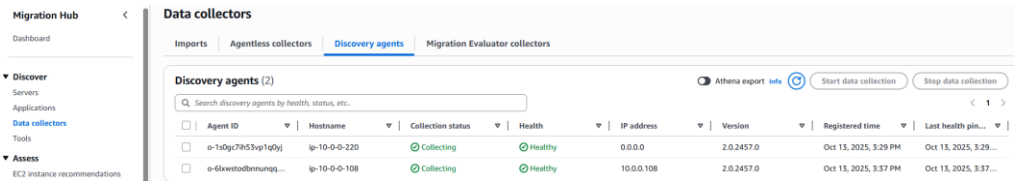
Install and deploy ADS agent on web server

```
$ sudo su - ubuntu
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

ubuntu@ip-10-0-0-220:~$ curl -o ./aws-discovery-agent.tar.gz https://s3-us-west-2.amazonaws.com/aws-discovery-agent.us-west-2/linux/latest/aws-discovery-agent.tar.gz
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100 26.0M  100 26.0M    0     0  76.5M      0 --:--:-- --:--:-- --:--:-- 76.8M
ubuntu@ip-10-0-0-220:~$ tar -xzf aws-discovery-agent.tar.gz
ubuntu@ip-10-0-0-220:~$ ll
total 53528
drwxr-xr-x 4 ubuntu ubuntu     4096 Oct 13 20:26 ./
drwxr-xr-x 4 root   root       4096 Oct 13 20:26 ../
-rw-r--r-- 1 ubuntu ubuntu      220 Apr  4  2018 .bash_logout
-rw-r--r-- 1 ubuntu ubuntu     3771 Apr  4  2018 .bashrc
drwx------ 3 ubuntu ubuntu     4096 Oct 13 20:26 .gnupg/
-rw-r--r-- 1 ubuntu ubuntu      807 Apr  4  2018 .profile
drwx------ 2 ubuntu ubuntu     4096 Oct 13 20:22 .ssh/
-rw-r--r-- 1 ubuntu ubuntu 13762922 Jul 31 21:00 aws-discovery-agent.deb
-rw-r--r-- 1 ubuntu ubuntu 13683751 Jul 31 21:00 aws-discovery-agent.rpm
-rw-rw-r-- 1 ubuntu ubuntu 27305514 Oct 13 20:26 aws-discovery-agent.tar.gz
-rw-r--r-- 1 ubuntu ubuntu    23960 Jul 31 21:00 install
ubuntu@ip-10-0-0-220:~$
ubuntu@ip-10-0-0-220:~$ sudo bash install -r us-west-2 -k AKIAVRQ7NXI6N2KDMCPG -s z1tu0AtC33hRGVVDczzFaYY0wnk1Pab6uGM80f9B
AWS region option specified as: us-west-2.
AWS key ID option specified as: AKIAVRQ7NXI6N2KDMCPG.
AWS key secret option specified.


****************************************************
    INIT PHASE
****************************************************
Distribution type of the machine is ubuntu.
Distribution of the machine is Ubuntu.
Revision of the distribution is 18.04.
There is no existing version of the AWS Discovery Agent on the machine.
Using deb and apt-get for package management. Expected agent package name is aws-discovery-agent.deb.
Installing using local file /home/ubuntu/aws-discovery-agent.deb


****************************************************
    INSTALL AGENT PHASE
****************************************************
Installing agent with dpkg...
dpkg: warning: overriding problem because --force enabled:
dpkg: warning: package architecture (i386) does not match system (amd64)
Selecting previously unselected package aws-discovery-agent:i386.
(Reading database ... 58669 files and directories currently installed.)
Preparing to unpack .../ubuntu/aws-discovery-agent.deb ...
Unpacking aws-discovery-agent:i386 (2457:2.0.2457.0-2457) ...
Setting up aws-discovery-agent:i386 (2457:2.0.2457.0-2457) ...
Created symlink /etc/systemd/system/multi-user.target.wants/aws-discovery-daemon.service → /usr/lib/systemd/system/aws-discovery-daemon.service.
Validating install arguments and writing configuration to /etc/opt/aws/discovery/config


****************************************************
    FINISH PHASE
****************************************************
ubuntu@ip-10-0-0-220:~$ sudo systemctl status aws-discovery-daemon.service
● aws-discovery-daemon.service - AWS Discovery Agent: Collects system configuration and usage statistics for AWS Discovery.
   Loaded: loaded (/usr/lib/systemd/system/aws-discovery-daemon.service; enabled; vendor preset: enabled)
   Active: active (running) since Mon 2025-10-13 20:28:12 UTC; 28s ago
 Main PID: 14935 (aws-discoveryd)
    Tasks: 8 (limit: 2317)
   CGroup: /system.slice/aws-discovery-daemon.service
           └─14935 /opt/aws/discovery/aws-discoveryd nodaemon

Oct 13 20:28:12 ip-10-0-0-220 systemd[1]: Started AWS Discovery Agent: Collects system configuration and usage statistics for AWS Discovery.
```

Install and deploy ADS agent on DB server

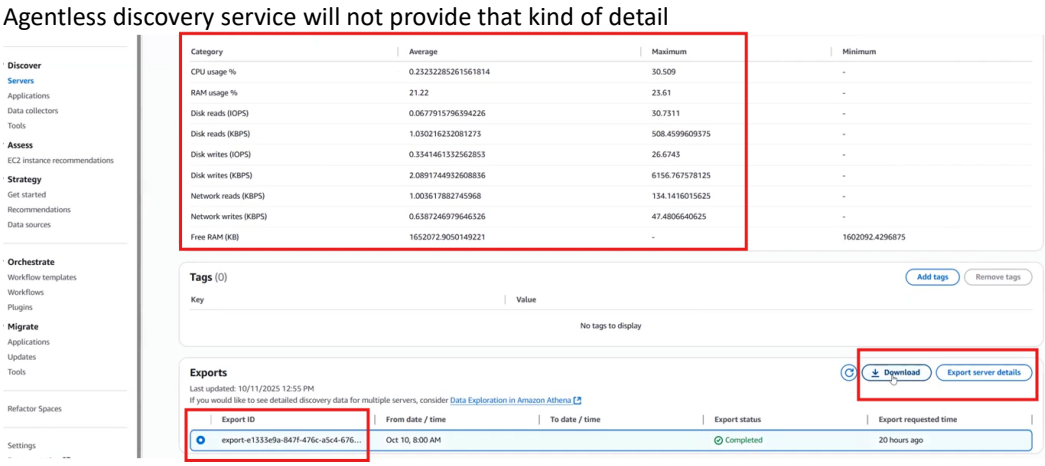
```
$ sudo su - ubuntu
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

ubuntu@ip-10-0-0-108:~$ curl -o ./aws-discovery-agent.tar.gz https://s3-us-west-2.amazonaws.com/aws-discovery-agent.us-west-2/linux/latest/aws-discovery-agent.tar.gz
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100 26.0M  100 26.0M    0     0  74.8M      0 --:--:-- --:--:-- --:--:-- 74.8M
ubuntu@ip-10-0-0-108:~$ tar -xzf aws-discovery-agent.tar.gz
ubuntu@ip-10-0-0-108:~$ ll
total 53528
drwxr-xr-x 4 ubuntu ubuntu     4096 Oct 13 20:35 ./
drwxr-xr-x 4 root   root       4096 Oct 13 20:35 ../
-rw-r--r-- 1 ubuntu ubuntu      220 Apr  4  2018 .bash_logout
-rw-r--r-- 1 ubuntu ubuntu     3771 Apr  4  2018 .bashrc
drwx------ 3 ubuntu ubuntu     4096 Oct 13 20:35 .gnupg/
-rw-r--r-- 1 ubuntu ubuntu      807 Apr  4  2018 .profile
drwx------ 2 ubuntu ubuntu     4096 Oct 13 20:22 .ssh/
-rw-r--r-- 1 ubuntu ubuntu 13762922 Jul 31 21:00 aws-discovery-agent.deb
-rw-r--r-- 1 ubuntu ubuntu 13683751 Jul 31 21:00 aws-discovery-agent.rpm
-rw-rw-r-- 1 ubuntu ubuntu 27305514 Oct 13 20:35 aws-discovery-agent.tar.gz
-rw-r--r-- 1 ubuntu ubuntu    23960 Jul 31 21:00 install
ubuntu@ip-10-0-0-108:~$
```

```
ubuntu@ip-10-0-0-108:~$ sudo bash install -r us-west-2 -k AKIAVRQ7NXI6N2KDMCPG -s zitu0Atc33hRGVVDczzFaYY0wnk1Pab6uGM8Of9B
AWS region option specified as: us-west-2.
AWS key ID option specified as: AKIAVRQ7NXI6N2KDMCPG.
AWS key secret option specified.


*****************************************************
    INIT PHASE
*****************************************************
Distribution type of the machine is ubuntu.
Distribution of the machine is Ubuntu.
Revision of the distribution is 18.04.
There is no existing version of the AWS Discovery Agent on the machine.
Using deb and apt-get for package management. Expected agent package name is aws-discovery-agent.deb.
Installing using local file /home/ubuntu/aws-discovery-agent.deb


*****************************************************
    INSTALL AGENT PHASE
*****************************************************
Installing agent with dpkg...
dpkg: warning: overriding problem because --force enabled:
dpkg: warning: package architecture (i386) does not match system (amd64)
Selecting previously unselected package aws-discovery-agent:i386.
(Reading database ... 57182 files and directories currently installed.)
Preparing to unpack .../ubuntu/aws-discovery-agent.deb ...
Unpacking aws-discovery-agent:i386 (2457:2.0.2457.0-2457) ...
Setting up aws-discovery-agent:i386 (2457:2.0.2457.0-2457) ...
Created symlink /etc/systemd/system/multi-user.target.wants/aws-discovery-daemon.service → /usr/lib/systemd/system/aws-discovery-daemon.service.
Validating install arguments and writing configuration to /etc/opt/aws/discovery/config


*****************************************************
    FINISH PHASE
*****************************************************
ubuntu@ip-10-0-0-108:~$ sudo systemctl status aws-discovery-daemon.service
● aws-discovery-daemon.service - AWS Discovery Agent: Collects system configuration and usage statistics for AWS Discovery.
   Loaded: loaded (/usr/lib/systemd/system/aws-discovery-daemon.service; enabled; vendor preset: enabled)
   Active: active (running) since Mon 2025-10-13 20:36:25 UTC; 33s ago
 Main PID: 3621 (aws-discoveryd)
    Tasks: 8 (limit: 1120)
   CGroup: /system.slice/aws-discovery-daemon.service
           └─3621 /opt/aws/discovery/aws-discoveryd nodaemon

Oct 13 20:36:25 ip-10-0-0-108 systemd[1]: Started AWS Discovery Agent: Collects system configuration and usage statistics for AWS Discovery..
```



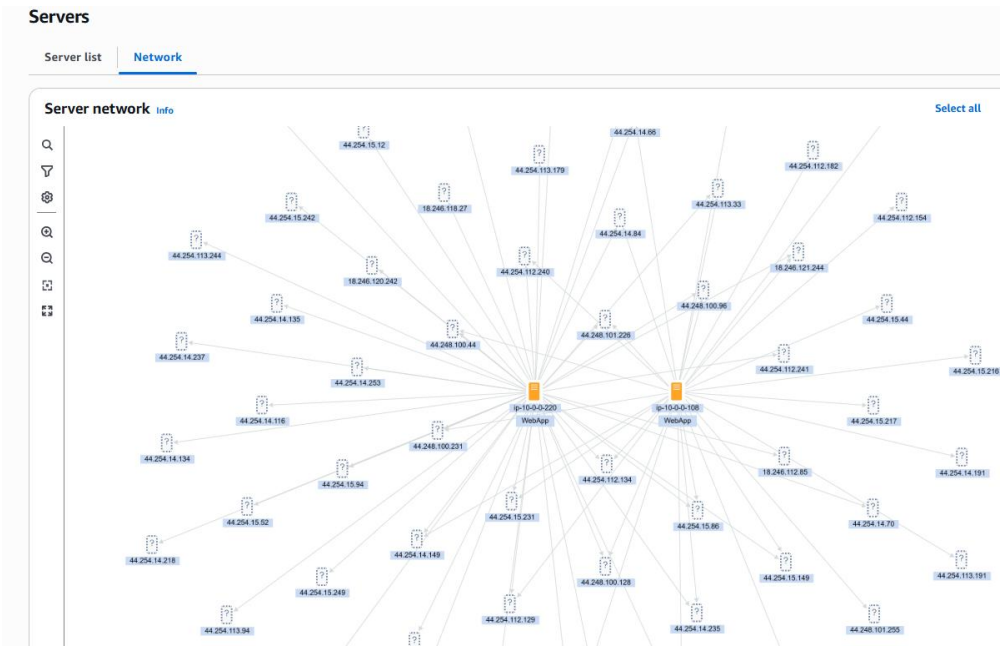
Check on the server one at a time

This will give you resource usage on that server and also downloading or exporting it will give info about all the process including 3rd party applications running on that server.
This is only possible if you do this with an agent the way we did.
Agentless discovery service will not provide that kind of detail

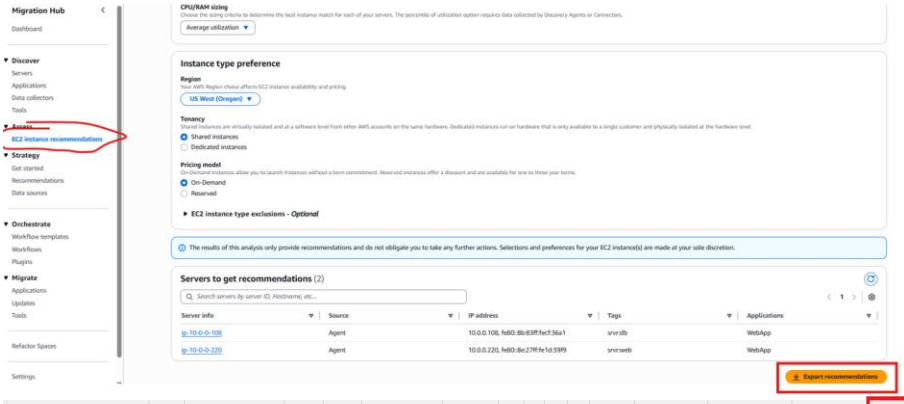


Running process details on the server

2.8E+11	o-4iyym4c D538E806I	FALSE	
2.8E+11	o-4iyym4c C18F551BI	FALSE	/usr/bin/dbus-daemon --system --address=systemd: --nofork --nopidfile --syste
2.8E+11	o-4iyym4c F368B79CI	FALSE	irqbalanc /usr/sbin/ /usr/sbin/irqbalance
2.8E+11	o-4iyym4c A1C2916EI	FALSE	atd /usr/sbin/ /usr/sbin/atd
2.8E+11	o-4iyym4c EEAD4ICZ	FALSE	/lib/systemd/systemd-logind
2.8E+11	o-4iyym4c 08F857F3I	FALSE	acpid /usr/sbin/ /usr/sbin/acpid
2.8E+11	o-4iyym4c 32D234E7I	FALSE	lxcfs /usr/bin/l: /usr/bin/lxcfs
2.8E+11	o-4iyym4c E9E2BD06-	FALSE	/sbin/agetty -o -p -- \u --keep-baud 115200,38400,9600 ttyS0 vt220
2.8E+11	o-4iyym4c CDF71E2B-	FALSE	/sbin/agetty -o -p -- \u --noclear tty1 linux
2.8E+11	o-4iyym4c 845AFDB3	FALSE	polkitd /usr/lib/p /usr/lib/policykit-1/polkitd
2.8E+11	o-4iyym4c 2955B943(	FALSE	/usr/bin/python3 /usr/share/unattended-upgrades/unattended-upgrade-shutd
2.8E+11	o-4iyym4c 6F1AC08F	FALSE	snapd /usr/lib/sr /snap/core/17247/usr/lib/snapd/snapd
2.8E+11	o-4iyym4c D538E806I	FALSE	
2.8E+11	o-4iyym4c D538E806I	FALSE	
2.8E+11	o-4iyym4c D538E806I	FALSE	
2.8E+11	o-4iyym4c D538E806I	FALSE	
2.8E+11	o-4iyym4c 9E5B9D6(	FALSE	/usr/bin/python3 /usr/bin/networkd-dispatcher --run-startup-triggers
2.8E+11	o-4iyym4c 386737CA:	FALSE	rsyslogd /usr/sbin/ /usr/sbin/rsyslogd
2.8E+11	o-4iyym4c 7A826323(	FALSE	uuidd /usr/sbin/ /usr/sbin/uuidd
2.8E+11	o-4iyym4c 9AC98723I	FALSE	accounts- /usr/lib/a /usr/lib/accountsservice/accounts-daemon
2.8E+11	o-4iyym4c 7D978BD0	FALSE	apache2 /usr/sbin/ /usr/sbin/apache2
2.8E+11	o-4iyym4c 98F32632(	FALSE	apache2 /usr/sbin/ /usr/sbin/apache2
2.8E+11	o-4iyym4c 98F32632(	FALSE	apache2 /usr/sbin/ /usr/sbin/apache2
2.8E+11	o-4iyym4c 98F32632(	FALSE	apache2 /usr/sbin/ /usr/sbin/apache2
2.8E+11	o-4iyym4c 98F32632(	FALSE	apache2 /usr/sbin/ /usr/sbin/apache2
2.8E+11	o-4iyym4c 98F32632(	FALSE	apache2 /usr/sbin/ /usr/sbin/apache2
2.8E+11	o-4iyym4c 98F32632(	FALSE	apache2 /usr/sbin/ /usr/sbin/apache2
2.8E+11	o-4iyym4c D538E806I	FALSE	
2.8E+11	o-4iyym4c D538E806I	FALSE	
2.8E+11	o-4iyym4c 98F32632(	FALSE	apache2 /usr/sbin/ /usr/sbin/apache2
2.8E+11	o-4iyym4c D538E806I	FALSE	
2.8E+11	o-4iyym4c 4FC48ADC	FALSE	amazon-s /snap/am /snap/amazon-ssm-agent/11797/amazon-ssm-agent
2.8E+11	o-4iyym4c ED A5857E(	FALSE	ssm-agen /snap/am /snap/amazon-ssm-agent/11797/ssm-agent-worker
2.8E+11	o-4iyym4c D538E806I	FALSE	
2.8E+11	o-4iyym4c D538E806I	FALSE	
2.8E+11	o-4iyym4c D538E806I	FALSE	
2.8E+11	o-4iyym4c D538E806I	FALSE	
2.8E+11	o-4iyym4c D538E806I	FALSE	
2.8E+11	o-4iyym4c D538E806I	FALSE	



RECOMMENDATIONS



ServerId	Server	Server.HostName	Server. Recom	Server.OS.Nam	Server.OS.	Serv	Ser	Serv	Rec	Recomm	Server.RAM.T	Recommendati	Recommenda	Recommendation.EC2.Instance.
d-server-036643j45txlcl		ip-10-0-0-220		Linux - Ubuntu	18.4.0.0	1	1	2	0.2	1	2048.917504	14.95247181	306.3638121	m5.large
d-server-0397sht93k00wv		ip-10-0-0-108		Linux - Ubuntu	18.4.0.0	1	1	2	0.2	1	1024.458752	44.39500161	454.8084794	m5.large

Completed part:

1. Initialized and configured AWS Application Discovery service (ADS).
    - Set Migration Hub home Region.
    - Created IAM user and credentials.
2. Deployed ADS Agent to the source **Webserver** and **DBServer**.
3. Verified that the data collection is started and in healthy condition.
4. Reviewed the data and the different options in the Migration Hub:
    - Server list.
    - Network visualization.
    - Application grouping.
    - Optionally create tags and assign them to the servers.
5. Worked with Application groups and tags:
    - Create application group.
    - Add or remove servers from the application group.
    - Optionally update the status of the application.

So now it's time to migrate and configure the database server!

Database Migration

Create following two SG

Security Groups (7) Info

Actions | Export secu

Find security groups by attribute or tag

Name	Security group ID	Security group name	VPC ID	Description
WebServerSG	sg-0700470f52c0fdaeb	WebServerSG	vpc-09955d99399f3493e	Web Server SG
RDS Target Database	sg-0de2a0fb1f7661f53	DB-SG	vpc-0dba11cbb944fb462	Security Group for RDS Target Databas
DMS replication Instance	sg-0f7c7f671ca28e66a	RI-SG	vpc-0dba11cbb944fb462	Security Group for DMS Replication Ins.
DBServerSG	sg-078fe07234ab86ddf	DBServerSG	vpc-09955d99399f3493e	DB Server SG

Create DB Subnet Group

Aurora and RDS > Subnet groups > Create DB subnet group

Create DB subnet group

To create a new subnet group, give it a name and a description, and choose an existing VPC. You will then be able to add subnets related to that VPC.

Subnet group details

Name
You won't be able to modify the name after your subnet group has been created.
database-subnet-group
Must contain from 1 to 255 characters. Alphanumeric characters, spaces, hyphens, underscores, and periods are allowed.

Description
Subnets where RDS will be deployed

VPC
Choose a VPC identifier that corresponds to the subnets you want to use for your DB subnet group. You won't be able to choose a different VPC identifier after your subnet group has been created.
cloud-app-vpc (vpc-0dba11cbb944fb462)
6 Subnets, 2 Availability Zones

Add subnets

Availability Zones
Choose the Availability Zones that include the subnets you want to add.
Choose an availability zone
us-west-2a | us-west-2b

Subnets
Choose the subnets that you want to add. The list includes the subnets in the selected Availability Zones.
Select subnets
cloud-app-vpc-private-a-db
Subnet ID: subnet-0eafef34dc57d1776 CIDR: 10.0.101.0/24
cloud-app-vpc-private-b-db
Subnet ID: subnet-0085382a8c8e8c719 CIDR: 10.0.201.0/24

For Multi-AZ DB clusters, you must select 3 subnets in 3 different Availability Zones.

Subnets selected (2)

Availability zone	Subnet name	Subnet ID	CIDR block
us-west-2a	cloud-app-vpc-private-a-db	subnet-0eafef34dc57d1776	10.0.101.0/24
us-west-2b	cloud-app-vpc-private-b-db	subnet-0085382a8c8e8c719	10.0.201.0/24

Create Target DB

Databases > cdr-cloud

cdr-cloud

Modify

Summary

DB identifier
cdr-cloud

CPU
34.03%

Status
Available

Class
db.t3.micro

Role
Instance

Current activity

Engine
MySQL Community

Region & AZ
us-west-2b

Recommendations

Connectivity & security | Monitoring | Logs & events | Configuration | Zero-ETL integrations | Maintenance & backups | Data migrations | Tags | Recommendations

Connectivity & security

Endpoint & port

Endpoint
cdr-cloud.cpq4caqu8n55.us-west-2.rds.amazonaws.com

Port
3306

Networking

Availability Zone
us-west-2b

VPC
cloud-app-vpc (vpc-0dba11cbb944fb462)

Subnet group
database-subnet-group

Subnets
subnet-0eafef34dc57d1776
subnet-0085382a8c8e8c719

Network type
IPv4

Security

VPC security groups
DB-SG (sg-0de2a0fb1f7661f53)
Active

Publicly accessible
No

Certificate authority Info
rds-ca-rsa2048-g1

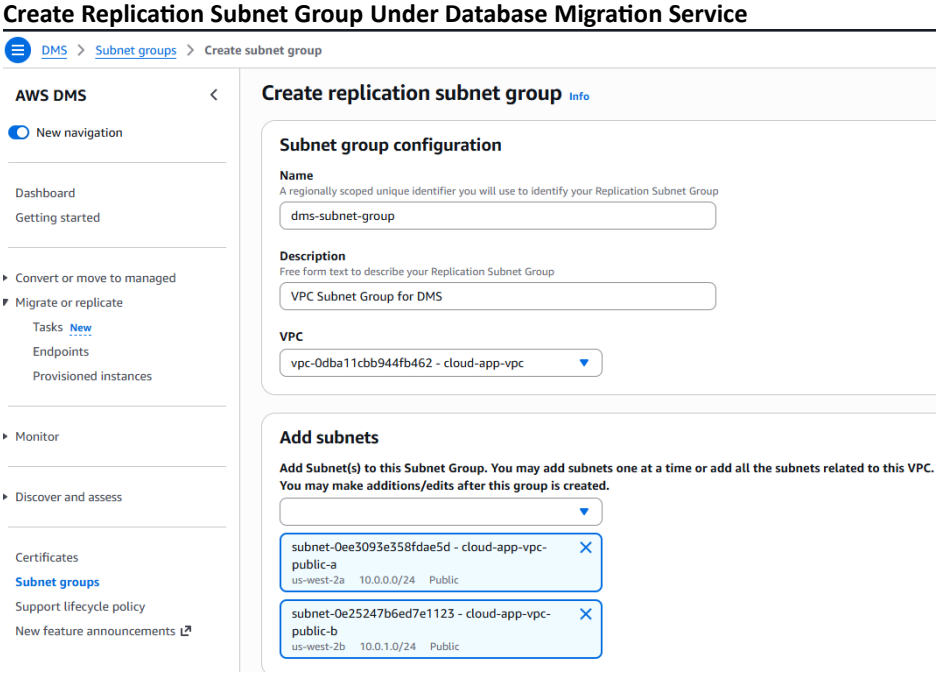
Certificate authority date
May 24, 2061, 17:59 (UTC-05:00)

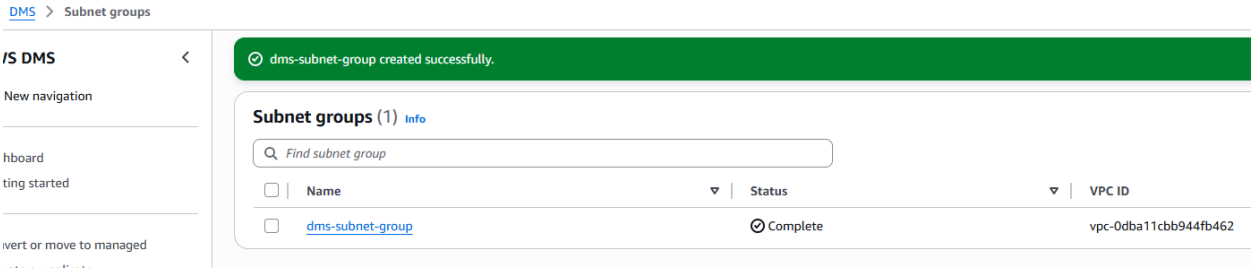
DB instance certificate expiration date
November 16, 2026, 10:35 (UTC-06:00)

Create Replication Instance

Now that we have the target database ready, it's time to prepare the resources needed to the migration with AWS Database Migration Server (DMS). One very important component is the Replication Instance, which is simply a managed Amazon Elastic Compute Cloud (Amazon EC2) instance that hosts one or more replication tasks. A single replication instance can host one or more replication tasks, depending on the characteristics of your migration and the capacity of the replication server.

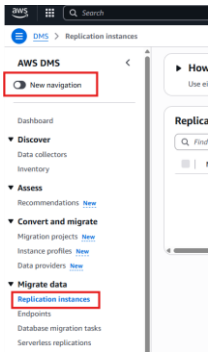
Create Replication Subnet Group Under Database Migration Service





Create AWS DMS Replication Instance

Now, we can go ahead and create the AWS Database Migration Service (DMS) Replication Instance that will be used to migrate the data from the source database to the managed target database on Amazon RDS.



DMS > Replication instances > Create replication instance

Settings

Name
The name must be unique among all of your replication instances in the current AWS region.

replication-instance

Replication instance name must not start with a numeric value

Descriptive Amazon Resource Name (ARN) - *optional*
A friendly name to override the default DMS ARN. You cannot modify it after creation.

Description - *optional*

DMS replication instance

The description must only have unicode letters, digits, whitespace, or one of these symbols: _.:/=+-@. 1000 maximum character.

Instance configuration Info

Instance class Info

dms.t3.medium
2 vCPUs 4 GiB Memory

Include previous-generation instance classes

Allocated storage (GiB) Info
Choose the amount of storage space you want for your replication instance. AWS DMS uses this storage for log files and cached t

50

Engine version
Choose an AWS DMS version to run on your replication instance. For more details, See the AWS DMS release notes. For inform

3.5.4

Include Beta DMS versions

High Availability Info
The Multi-AZ option deploys a primary replication instance in one Availability Zone (AZ) and a standby in another AZ. The Single-

Dev or test workload (Single-AZ)

Connectivity and security Info

Network type - new Info
To use dual-stack mode, make sure that you associate an IPv6 CIDR block with a subnet in the VPC you specify.

- IPv4
  Replication instance with an IPv4 network type that supports IPv4 addressing.
- Dual-stack mode
  Replication instance with addressing.

Virtual private cloud (VPC) for IPv4 Info
Choose the VPC where you want your replication instances to run. It includes VPCs in IPv4 and dual-stack mode.

cloud-app-vpc (vpc-0dba11cbb944fb462)

Create a new VPC.

Replication subnet group
Choose a subnet group for your replication instance. The subnet group defines the IP ranges and subnets that your replication in

dms-subnet-group

Public accessible
If you choose this option, AWS DMS will assign a public IP address to your replication instance, and you'll be able to connect

Advanced settings

Availability zone
Choose an availability zone (AZ) where you want your replication instance to run. The default is "No preference", meaning that A

us-west-2a

VPC security groups
Choose one or more security groups for your replication instances. The security groups specify inbound and outbound rules to co

Choose VPC to select associated VPC security group

RI-SG
Security Group for DMS Replication Instance

AWS KMS key Info

aws/dms

Replication instances (1)

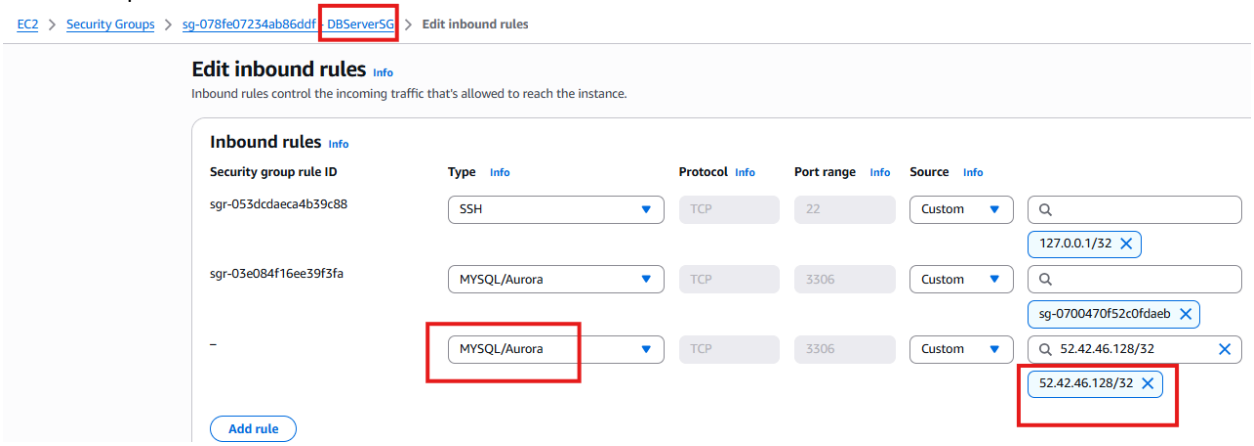
Actions | Create replication instance

Find replication instance

1

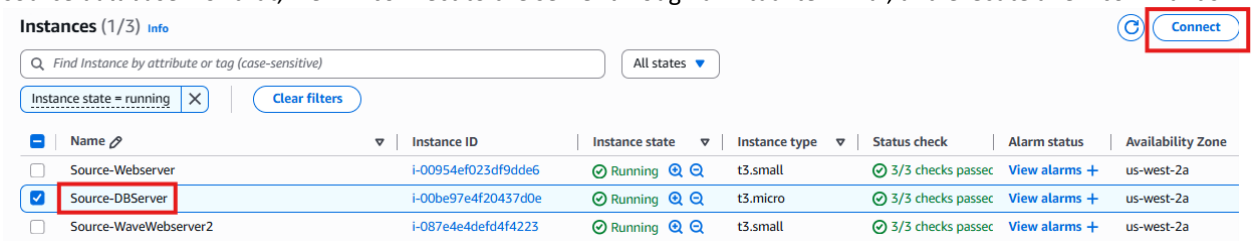
Name	Status	VPC	Class	Engine ...	Availab...	Networ...	Public	Public IP address	Private...	Multi AZ	Created
replicatio...	Available	vpc-0dba...	dms.t3.m...	3.5.4	us-west-2a	IPv4	Yes	52.42.46.128	10.0.0.5	No	November...

Add new replication instance into inbound rule for DBServerAG



Enable CDC on source database

For AWS DMS continuous replication with MySQL database as a source, we need to adjust the configuration of the source database. For that, we will connect to the server through a virtual terminal, and execute a few commands



In the terminal, we need connect to MySQL server and grant privileges for to the database user (the user we will use with the CDC feature):

sudo mysql -u root -pAWSRocksSince2006

*GRANT REPLICATION CLIENT ON *.* to '**your_db_user_name**';*
*GRANT REPLICATION SLAVE ON *.* to '**your_db_user_name**';*
*GRANT SUPER ON *.* to ' **your_db_user_name**';*

exit

```
$ sudo mysql -u root -pAWSRocksSince2006

GRANT REPLICATION CLIENT ON *.* to 'wordpress-user';
GRANT REPLICATION SLAVE ON *.* to 'wordpress-user';
GRANT SUPER ON *.* to 'wordpress-user';

exitmysql: [Warning] Using a password on the command line interface can be insecure.
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 17
Server version: 5.7.42-0ubuntu0.18.04.1 (Ubuntu)

Copyright (c) 2000, 2023, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql>
mysql> GRANT REPLICATION CLIENT ON *.* to 'wordpress-user';
Query OK, 0 rows affected (0.00 sec)

mysql> GRANT REPLICATION SLAVE ON *.* to 'wordpress-user';
Query OK, 0 rows affected (0.00 sec)

mysql> GRANT SUPER ON *.* to 'wordpress-user';
Query OK, 0 rows affected (0.00 sec)

mysql>
mysql> exit
Bye
$
```

In MySQL, the binary log contains “events” that describe database changes such as table creation operations or changes to table data. This is crucial for the continuous replication task with DMS. In the same terminal, let's create the directory structure for MySQL binary logs:

sudo su -
mkdir /var/lib/mysql/binlogs
chown -R mysql\:mysql /var/lib/mysql/binlogs
ls -ld /var/lib/mysql/binlogs
exit

```
$ sudo su -
root@ip-10-0-0-155:~# mkdir /var/lib/mysql/binlogs
root@ip-10-0-0-155:~# chown -R mysql\:mysql /var/lib/mysql/binlogs
root@ip-10-0-0-155:~# ls -ld /var/lib/mysql/binlogs
drwxr-xr-x 2 mysql mysql 4096 Nov 16 18:31 /var/lib/mysql/binlogs
root@ip-10-0-0-155:~# exit
logout
$
```

Modify the MySQL configuration to enable the Binary Log. In the terminal, execute:

sudo su -
cp /etc/mysql/mysql.conf.d/mysqld.cnf /etc/mysql/my.cnf
chown -R mysql\:mysql /etc/mysql/my.cnf
cat <<EOF >> /etc/mysql/my.cnf
server_id=1
log-bin=/var/lib/mysql/binlogs/log
binlog_format=ROW
expire_logs_days=1
binlog_checksum=NONE
binlog_row_image=FULL

log_slave_updates=TRUE
performance_schema=ON
EOF

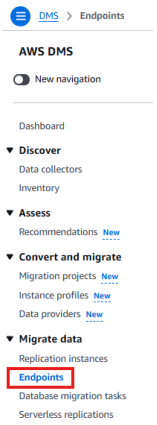
```
$ sudo su -
root@ip-10-0-0-155:~# sudo su -
root@ip-10-0-0-155:~# cp /etc/mysql/mysql.conf.d/mysqld.cnf /etc/mysql/my.cnf
root@ip-10-0-0-155:~# chown -R mysql\:mysql /etc/mysql/my.cnf
root@ip-10-0-0-155:~# cat <<EOF >> /etc/mysql/my.cnf
> server_id=1
> log-bin=/var/lib/mysql/binlogs/log
> binlog_format=ROW
> expire_logs_days=1
> binlog_checksum=NONE
> binlog_row_image=FULL
> log_slave_updates=TRUE
> performance_schema=ON
> EOF
```

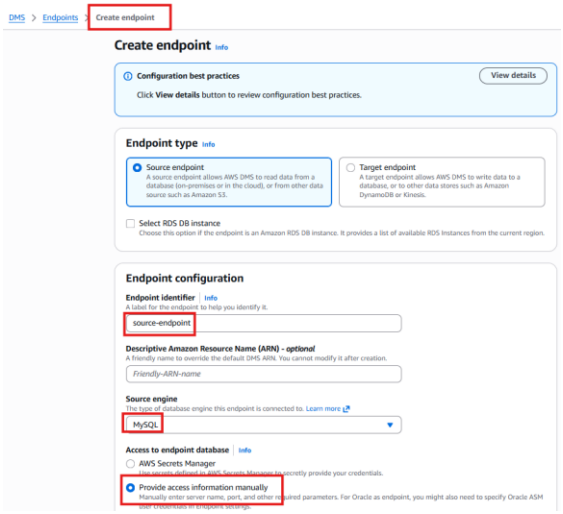
Verify if the changes took effect running the commands and comparing with the screenshot

```
root@ip-10-0-0-155:~# sudo systemctl restart mysql
root@ip-10-0-0-155:~# sudo mysql -u root -pAWSRocksSince2006
mysql: [Warning] Using a password on the command line interface can be insecure.
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 2
Server version: 5.7.42-0ubuntu0.18.04.1-log (Ubuntu)

Copyright (c) 2000, 2023, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> select variable_value as "BINARY LOGGING STATUS (log-bin) :: "
    -> from performance_schema.global_variables where variable_name='log_bin';
+-------------------------------------+
| BINARY LOGGING STATUS (log-bin) ::  |
+-------------------------------------+
| ON                                  |
+-------------------------------------+
1 row in set (0.01 sec)

mysql>
mysql>
mysql> select variable_value as "BINARY LOG FORMAT (binlog_format) :: "
    -> from performance_schema.global_variables where variable_name='binlog_format';
+---------------------------------------+
| BINARY LOG FORMAT (binlog_format) ::  |
+---------------------------------------+
| ROW                                   |
+---------------------------------------+
1 row in set (0.00 sec)
```

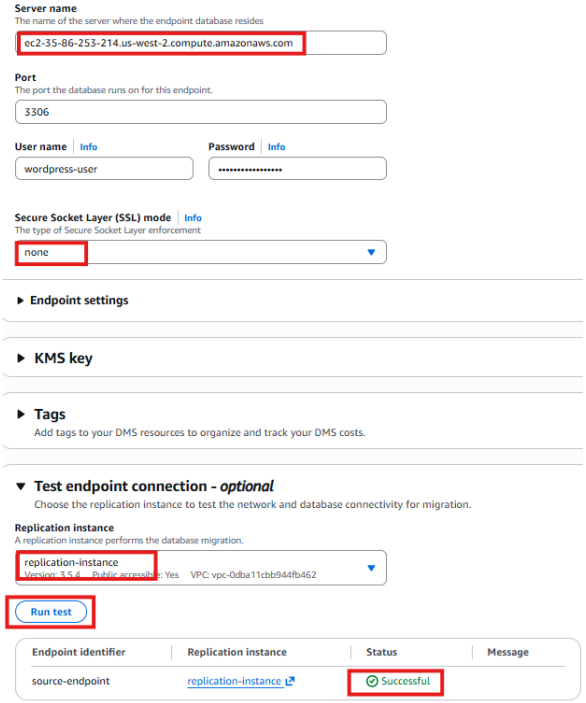
Create source and target endpoints

AWS DMS uses an endpoint to access your source or target data store, to execute the replication tasks. Let's create specific endpoints to our source database server, running in the EC2 Instance named Source-DBServer, and our target managed database, running on Amazon RDS.





Get 'DBServerDNSName' from cloudformation Resource tab and add here for **Server Name**



Name	Type	Status	Engine	Server name
source-endpoint	Source	Active	MySQL	ec2-35-86-253-214.us-west-2.compute.amazonaws.com

Create Target endpoint

DMS > Endpoints > Create endpoint

Endpoint type Info

- Source endpoint
  A source endpoint allows AWS DMS to read data from a database (on-premises or in the cloud), or from other data source such as Amazon S3.
- Target endpoint
  A target endpoint allows AWS DMS to write data to a database, or to other data stores such as Amazon DynamoDB or Kinesis.

☑ Select RDS DB instance
Choose this option if the endpoint is an Amazon RDS DB instance. It provides a list of available RDS Instances from the current region.

RDS Instance
Instances available only for current user and region
cdr-cloud

Endpoint configuration

Endpoint identifier Info
A label for the endpoint to help you identify it.
cdr-cloud

Descriptive Amazon Resource Name (ARN) - *optional*
A friendly name to override the default DMS ARN. You cannot modify it after creation.
Friendly-ARN-name

Target engine
The type of database engine this endpoint is connected to. Learn more
MySQL

DMS > Endpoints > Create endpoint

Access to endpoint database Info

- AWS Secrets Manager
  Use secrets defined in AWS Secrets Manager to secretly provide your credentials.
- Provide access information manually
  Manually enter server name, port, and other required parameters. For Oracle as endpoint, you might also need to specify Oracle ASM user credentials in Endpoint settings.
- IAM authentication - *new*
  Use existing or create new IAM roles to manage endpoint access. Available only for RDS databases. Learn more

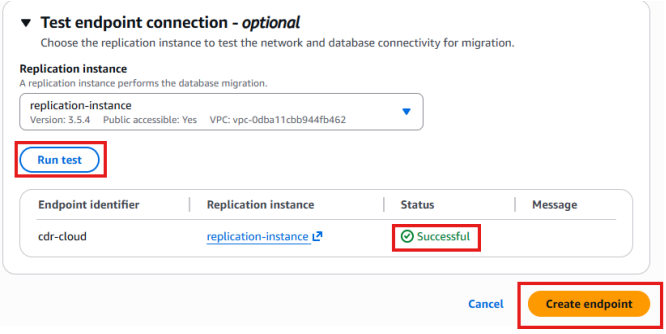
Server name
The name of the server where the endpoint database resides
cdr-cloud.cpq4caqu8n55.us-west-2.rds.amazonaws.com

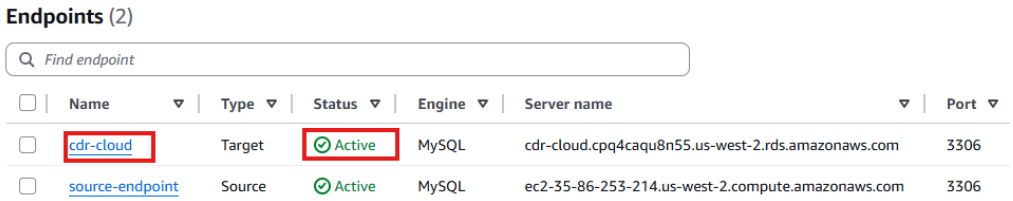
Port
The port the database runs on for this endpoint.
3306

User name Info
admin

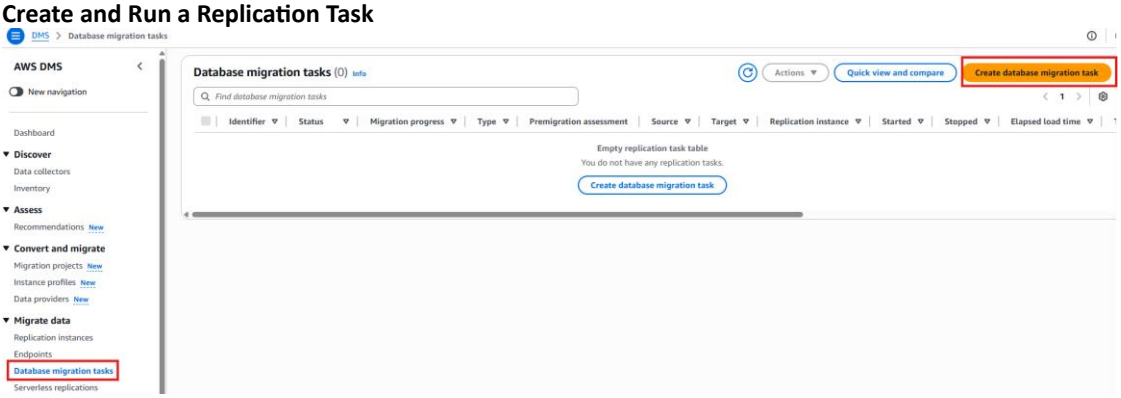
Password Info
••••••••••••••••

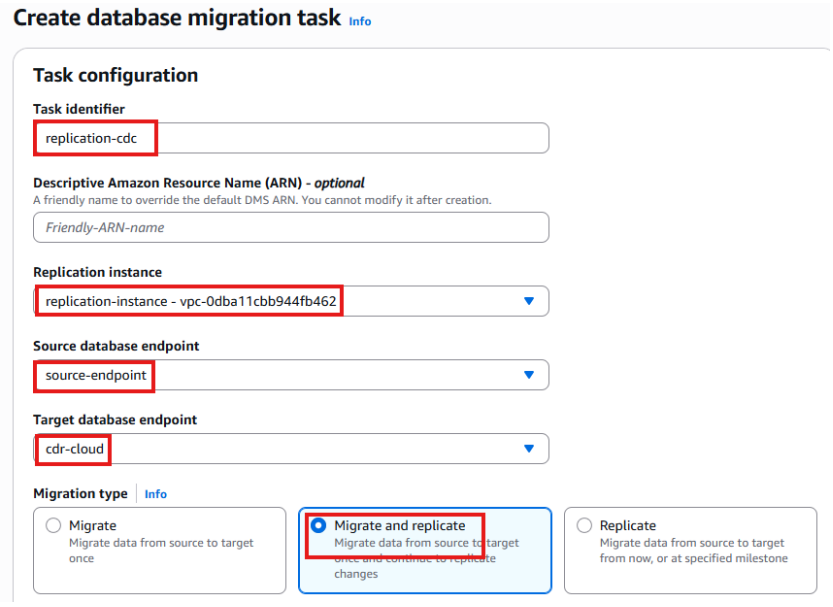
Secure Socket Layer (SSL) mode Info
The type of Secure Socket Layer enforcement
none





Create and Run a Replication Task





Task settings

Editing mode | Info

- (●) Wizard
  You can enter only a subset of the available task settings.
- ( ) JSON editor
  You can enter all available task settings directly in JSON format.

Custom CDC stop mode for source transactions | Info

- (●) Disable custom CDC stop mode
- ( ) Enable custom CDC stop mode

Create recovery table on target DB | Info

- [ ] Turn on

Target table preparation mode | Info

- (●) Do nothing
- ( ) Drop tables on target
- ( ) Truncate

Stop task after full load completes | Info

- (●) Do not stop
- ( ) Stop before applying cached changes
- ( ) Stop after applying cached changes

LOB column settings | Info

- ( ) Do not include LOB columns
- ( ) Full LOB mode
- (●) Limited LOB mode

Data Validation | Info

- ( ) Turn off
  Do not validate data for the table in migration task.
- (●) Validation with data migration
  Choose this setting if you want AWS DMS to compare the data at the source and the target immediately after it performs a full data load. Validation ensures that DMS migrated your data accurately, but it requires additional time to complete.
- ( ) Validation without data migration (Not supported for Full load and CDC task)
  Choose this setting if you want your task to perform data validation without performing any migration or replication of data. You must set **Target table preparation mode** to **Do nothing**. This setting can only apply to **Migrate existing data (Full Load)** or **Replicate ongoing changes (CDC)** tasks. You cannot modify the validation-only setting after you create the task.

Task logs | Info

- [x] Turn on CloudWatch logs
  DMS task logging uses Amazon CloudWatch to log information during the migration process. You can change the component activities logged and the amount of information logged for each one.

ⓘ CloudWatch logs usage is charged at standard rates. For more details, see Amazon CloudWatch Pricing ↗.

Log context | Info

- [x] Turn on log context
  Select this setting if you want DMS to add context information to the task logs, which can make troubleshooting easier.

Table mappings Info

Editing mode

- (●) Wizard
  You can enter only a subset of the available table mappings.
- ( ) JSON editor
  You can enter all available table mappings directly in JSON format.

Specify at least one selection rule with an include action. After you do this, you can add one or more transformation rules.

▼ **Selection rules**

Choose the schema and/or tables you want to include with, or exclude from, your migration task. [Add new selection rule]

▼ where **schema name** is like 'wordpress-db' and **Source table name** is like '%', include

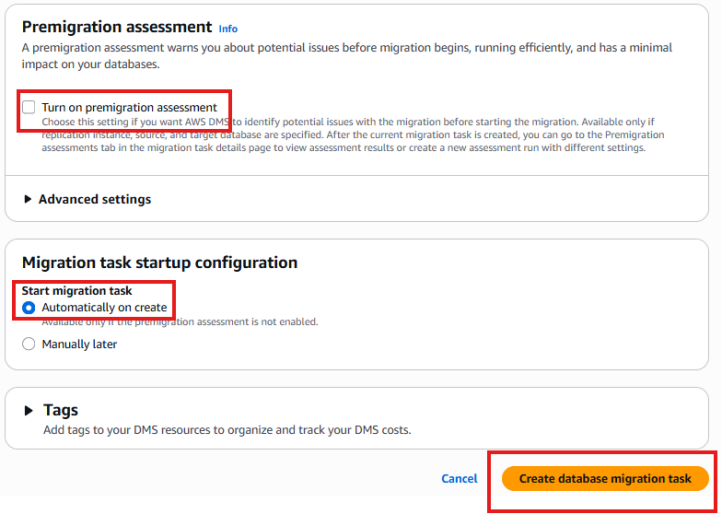
Schema

Enter a schema

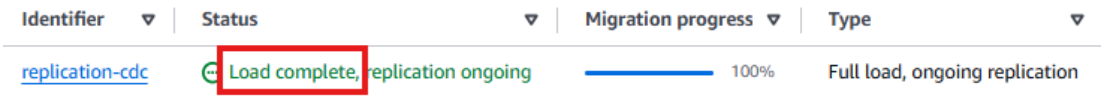
Source name

Use the % character as a wildcard

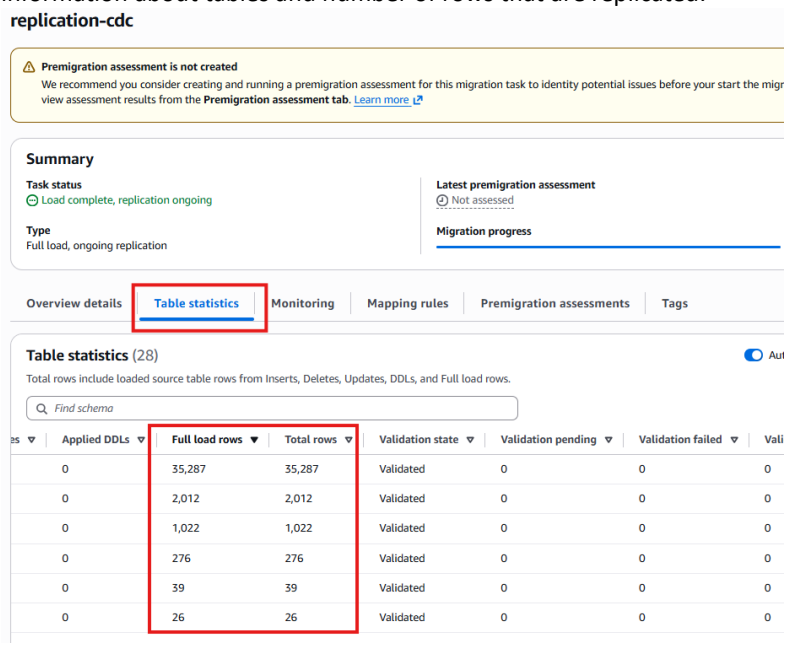
wordpress-db



Monitor the task until the status is changed to **Load complete**, replication ongoing and the progress is **100%**



This task will be running until we stop it, because it is a continuous replication. You can review its details to see information about tables and number of rows that are replicated:



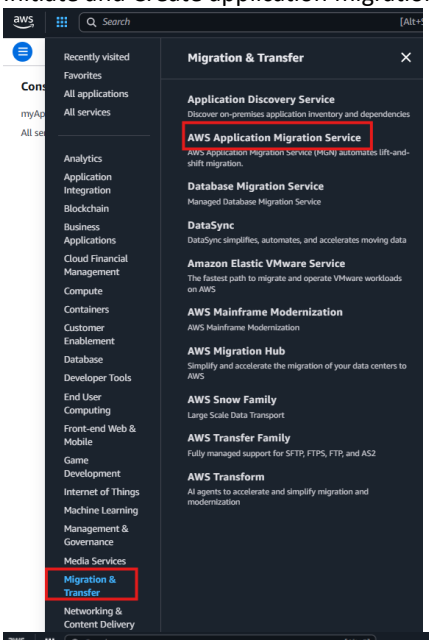
Work done:

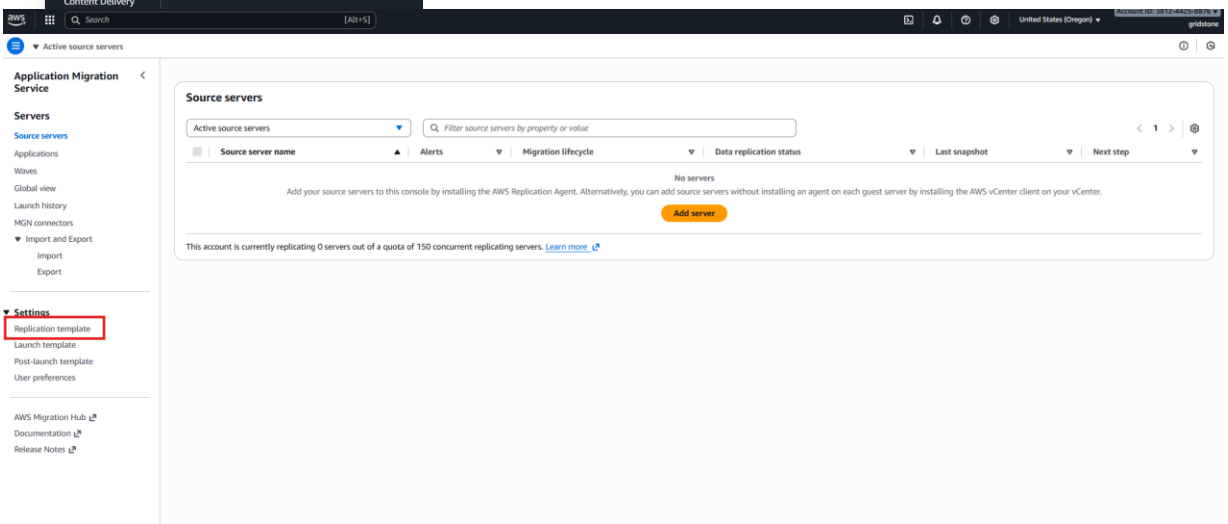
1. Created a new managed database, using Amazon Relational Database Service (RDS)
2. Created AWS Database Migration Service (DMS) Replication Instance - that allows you to replicate data between databases
3. Created the source and target DMS Endpoints
4. Modified the configuration of the source database to allow for continuous replication of data (binary log)
5. Started the replication of data through a DMS replication task

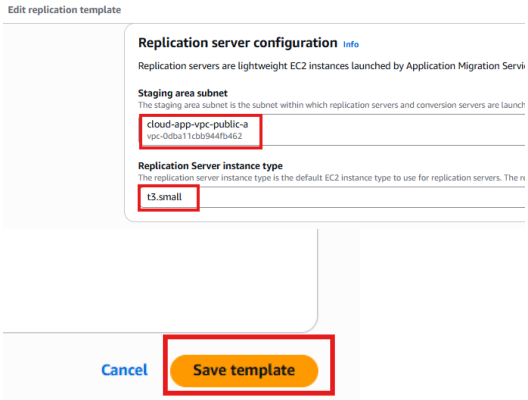
Now, our target environment is like this:

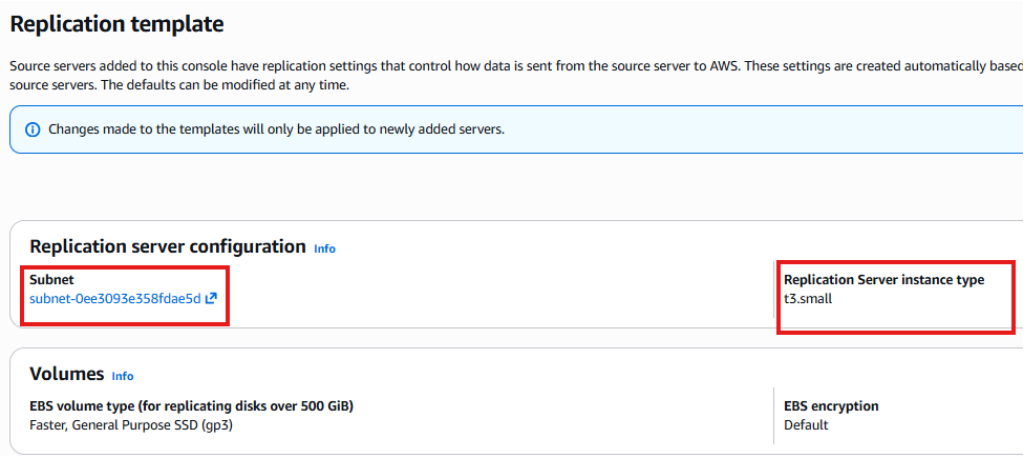
Application Migration

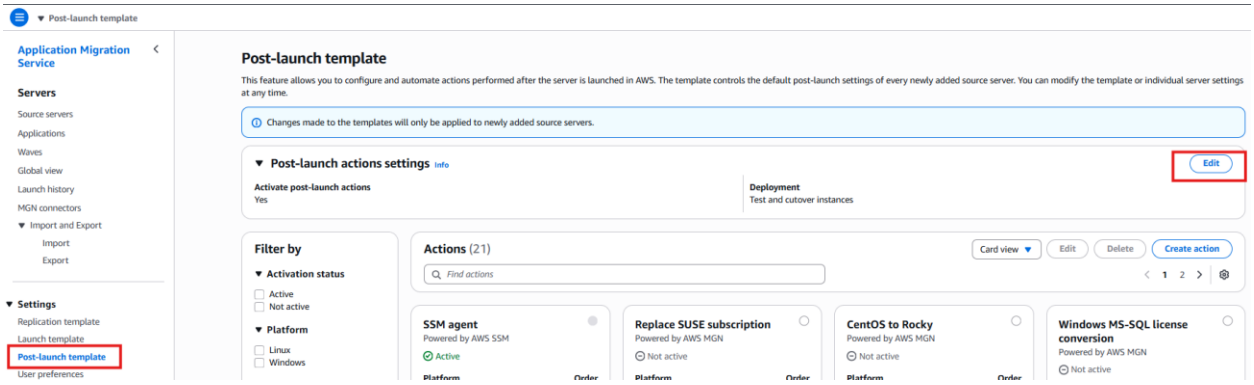
Initiate and Create application migration service template

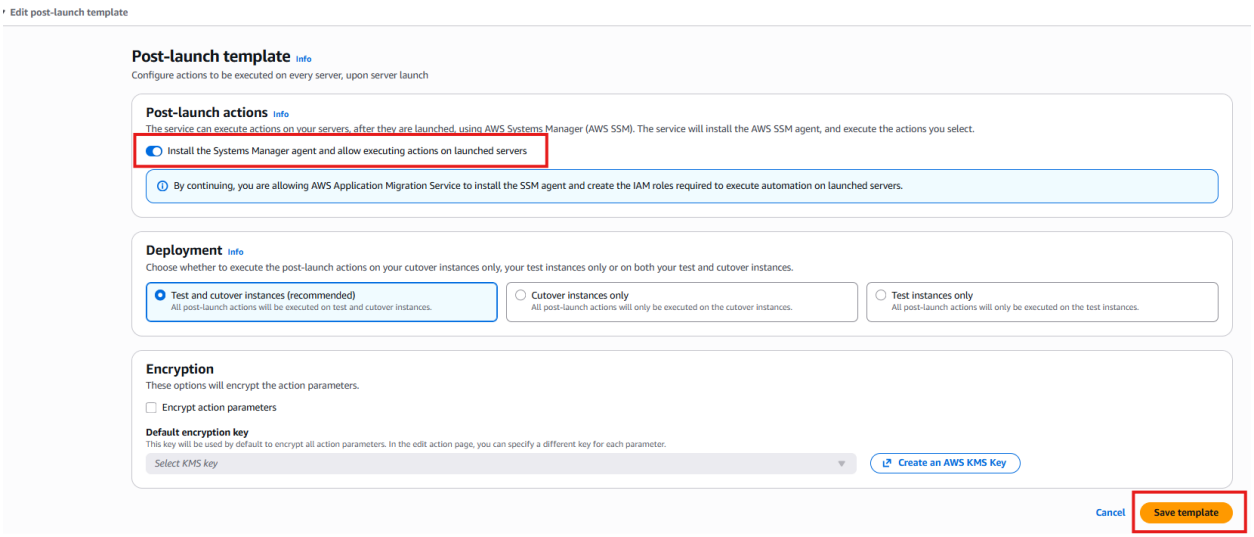












Install and Execute Agent on Source Web/Application Server

For this use case, Application Migration Service (MGN) will rehost the servers to AWS by replicating block-level data from the source target to the cloud. This can be agent-based or agent-less and we will rely on the agent-based replication for this use case, which is the default recommendation for most use cases.

The AWS Replication Agent performs an initial block-level read of the content of any volume attached to the server and replicates it to the replication server. The agent then acts as an OS-level read filter to capture writes and synchronizes any block level modifications to the AWS Application Migration Service replication server, ensuring near-zero RPO.

Grab app. migration service access key and secret from cloudformation output

Stack info | Events | Resources | Outputs | Parameters | Template | Change sets

Outputs (9)

Search outputs

Key	Value
ADSServiceAccessKey	AKIAVRQ7NXI6KSBZA4MP
ADSServiceSecretKey	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
AppMigServiceAccessKey	AKIAVRQ7NXI6GIRHIFUC
AppMigServiceSecretAccessKey	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
DBServerDNSName	ec2-35-86-253-214.us-west-2.compute.amazonaws.com

Connect to source web server

Instances (1/3) Info

Connect

Find Instance by attribute or tag (case-sensitive) | All states

Instance state = running | Clear filters

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone
Source-Webserver	i-00954ef023df9dde6	Running	t3.small	3/3 checks passed	View alarms +	us-west-2a
Source-DBServer	i-00be97e4f20437d0e	Running	t3.micro	3/3 checks passed	View alarms +	us-west-2a

Connect Info

Connect to an instance using the browser-based client.

EC2 Instance Connect | Session Manager | SSH client | EC2 serial console

Introducing Systems Manager just-in-time node access
Move towards zero standing privileges by requiring operators to request access before remotely connecting to instances. Learn more

Session Manager usage:

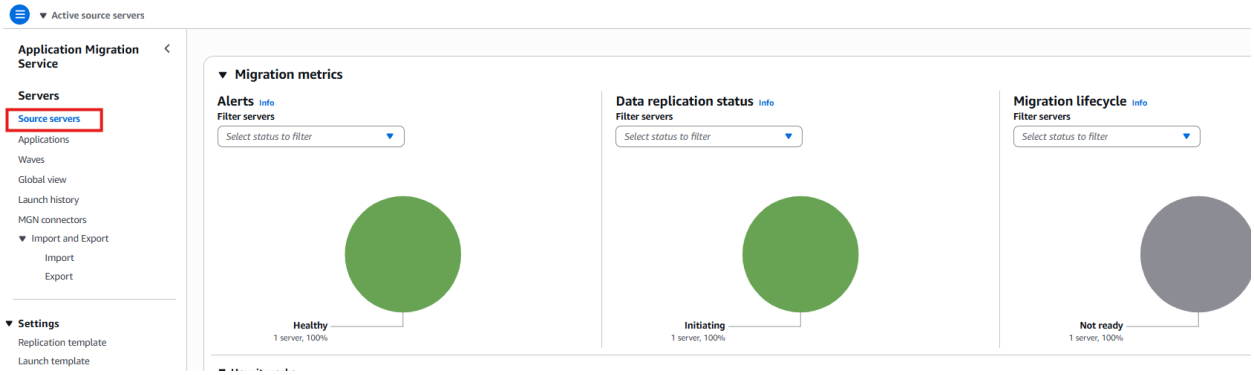
- Connect to your instance without SSH keys, a bastion host, or opening any inbound ports.
- Sessions are secured using an AWS Key Management Service key.
- You can log session commands and details in an Amazon S3 bucket or CloudWatch Logs log group.
- Configure sessions on the Session Manager Preferences page.

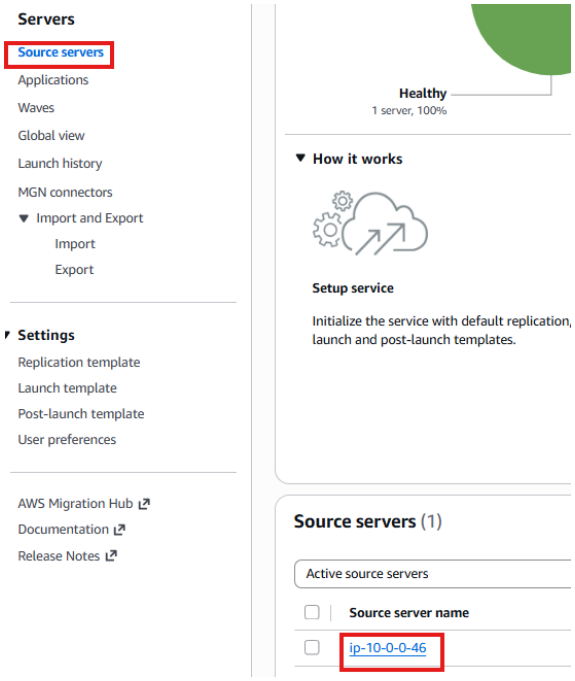
we need to download the MGN Agent installation script and execute it. For that, run the following commands:

```
$ cd ~
$ wget -O ./aws-replication-installer-init.py https://aws-application-migration-service-us-west-2.s3.amazonaws.com/latest/linux/aws-replication-installer-init.py
--2025-11-16 21:30:26--  https://aws-application-migration-service-us-west-2.s3.amazonaws.com/latest/linux/aws-replication-installer-init.py
Resolving aws-application-migration-service-us-west-2.s3.amazonaws.com (aws-application-migration-service-us-west-2.s3.amazonaws.com)... 52.218.253.171, 52.92.154.41, 52.92
Connecting to aws-application-migration-service-us-west-2.s3.amazonaws.com (aws-application-migration-service-us-west-2.s3.amazonaws.com)|52.218.253.171|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 31750 (31K) [binary/octet-stream]
Saving to: ‘./aws-replication-installer-init.py’

./aws-replication-installer-init.py                 100%[===================================================================================================

2025-11-16 21:30:26 (24.6 MB/s) - ‘./aws-replication-installer-init.py’ saved [31750/31750]
```

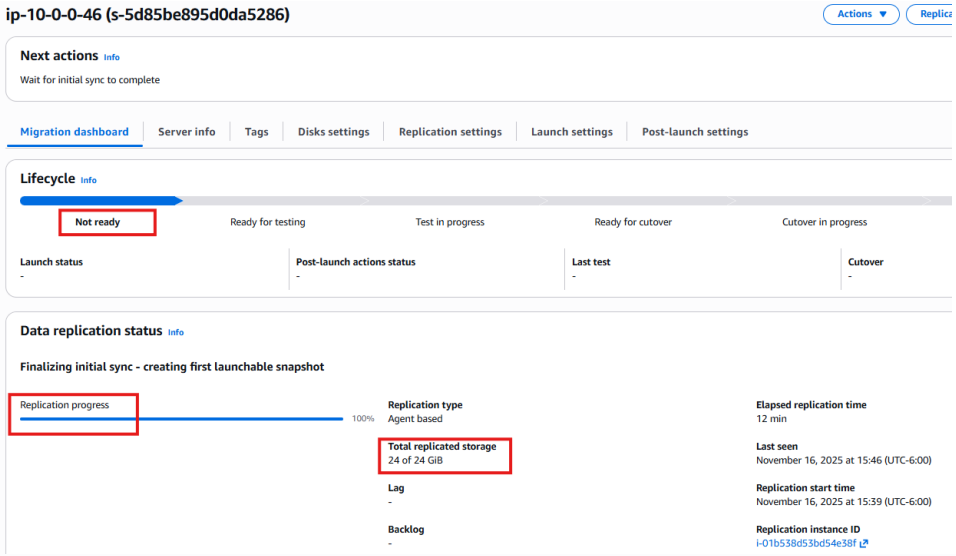
```
$ sudo python3 aws-replication-installer-init.py
The installation of the AWS Replication Agent has started.
AWS Region Name: us-west-2
AWS Access Key ID: AKIAVRQ7NXI6GIRHIFUC
AWS Secret Access Key: ****************************************
Identifying volumes for replication.
Choose the disks you want to replicate. Your disks are: /dev/nvme1n1,/dev/nvme0n1
To replicate some of the disks, type the path of the disks, separated with a comma (for example, /dev/sda,/dev/sdb). To replicate all disks, press Enter:
Identified volume for replication: /dev/nvme1n1 of size 8 GiB
Identified volume for replication: /dev/nvme0n1 of size 16 GiB
All volumes for replication were successfully identified.
Downloading the AWS Replication Agent onto the source server...
Finished.
Installing the AWS Replication Agent onto the source server...
Finished.
Syncing the source server with the Application Migration Service Console...
Finished.
The following is the source server ID: s-5d85be895d0da5286.
You now have 1 active source server out of a total quota of 150.
Learn more about increasing source servers limit at https://docs.aws.amazon.com/mgn/latest/ug/MGN-service-limits.html
The AWS Replication Agent was successfully installed.
$
```

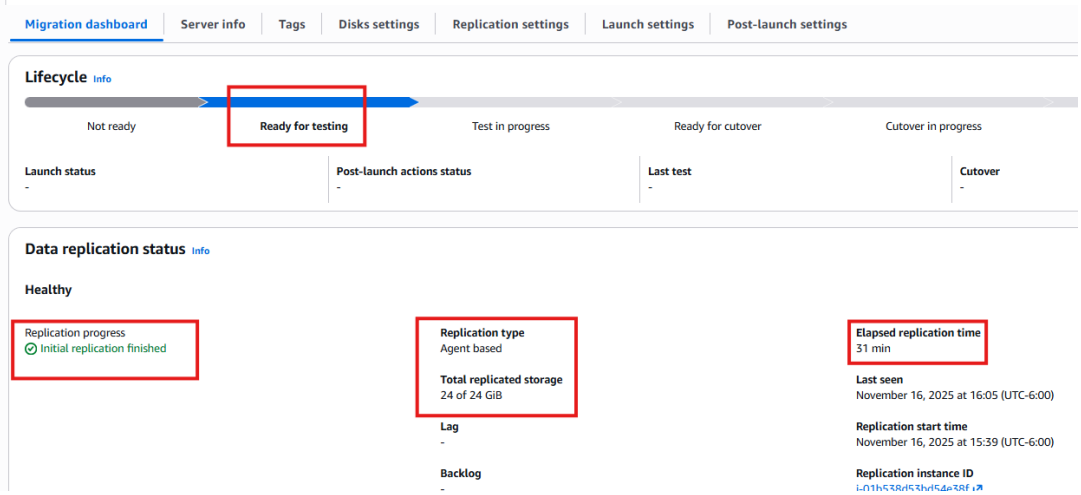
Once the agent installation finishes, you can go to AWS Console and access Services > Migration & Transfer > Application Migration Service. Then, you should land on Source servers page and see your source server doing its initial synchronization.



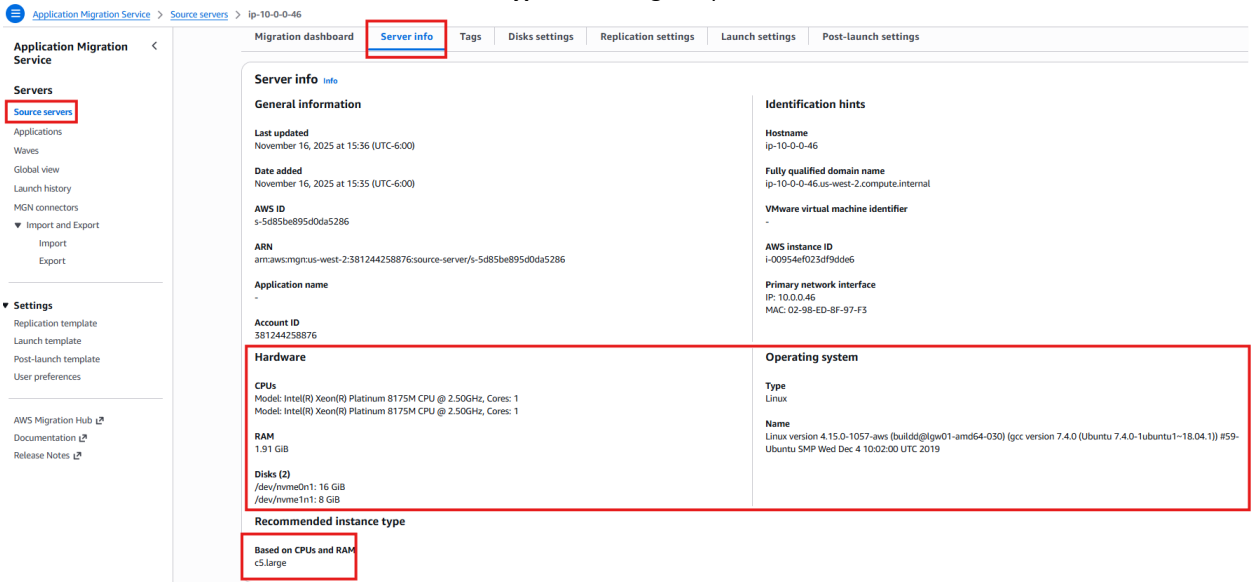


Check migration status

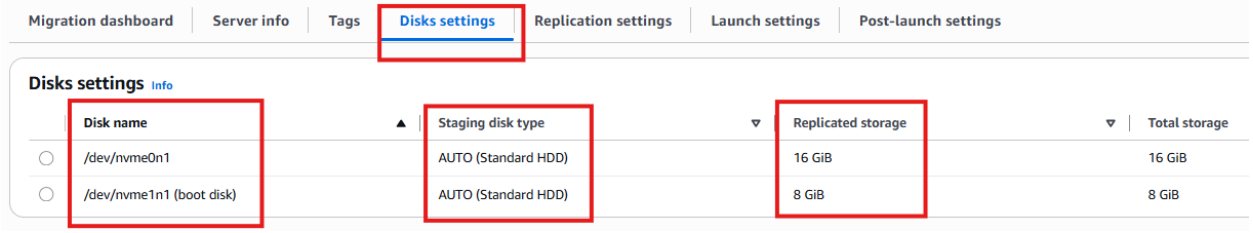




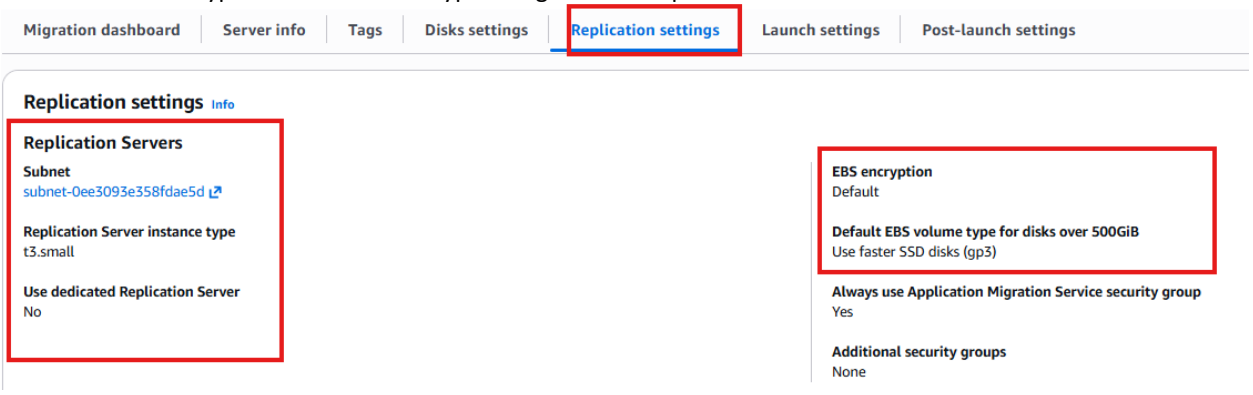
At Server info tab, you can review the server specifications, such as hardware profile and operating system, and also the **recommended Amazon EC2 instance type** according the profile of the source server



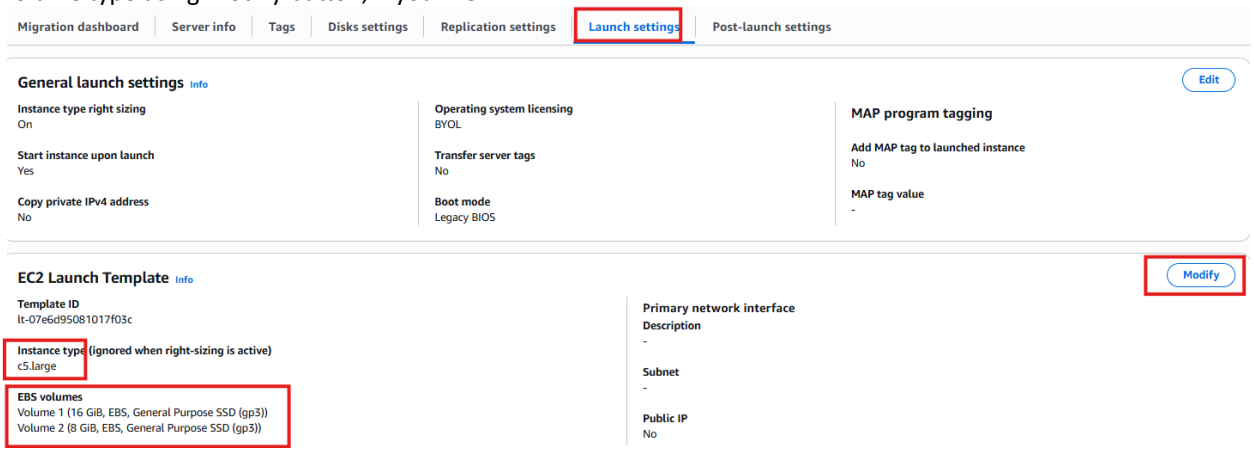
At Disk settings tab, you can see which disks were identified on source server and replication and disk type used in staging area:



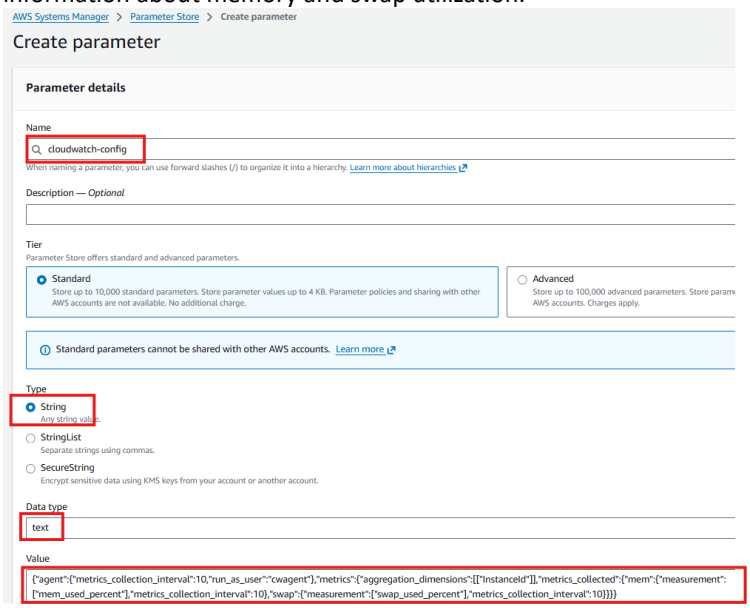
This shows server type and disk volume type being used for replication instance

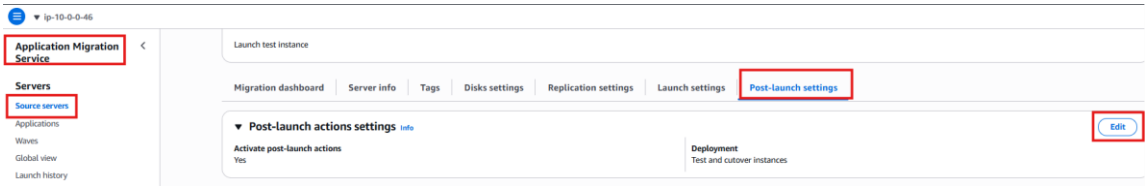


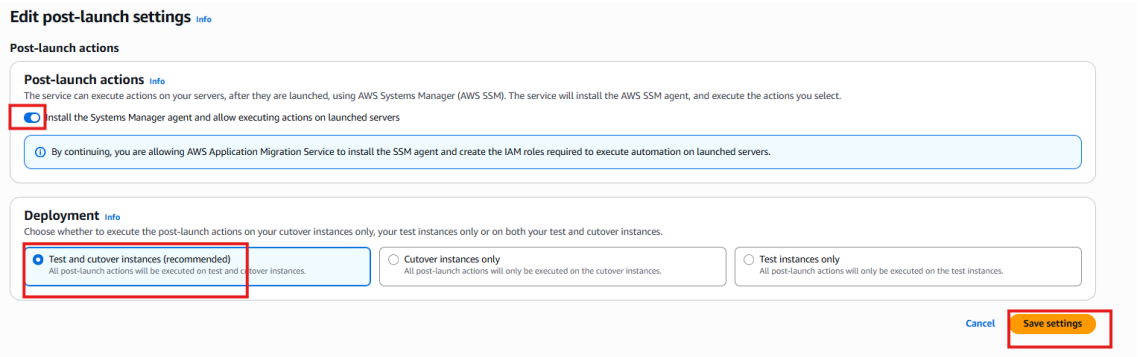
This tab shows the ec2 and disk volume type for destination server after migration. You can the instance or ebs volume type using Modify button, if you like

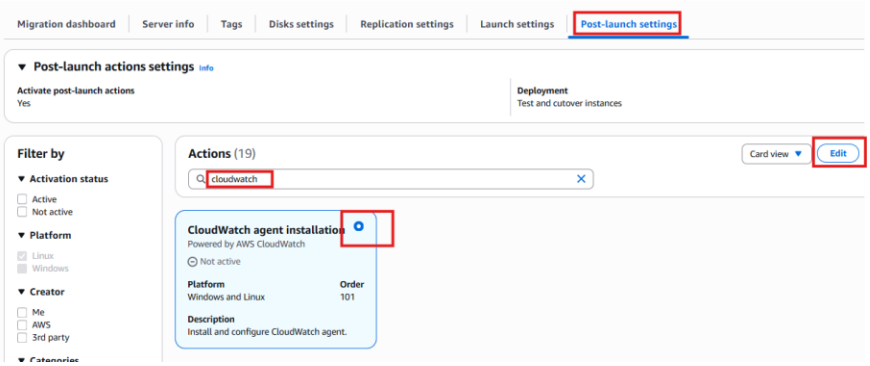


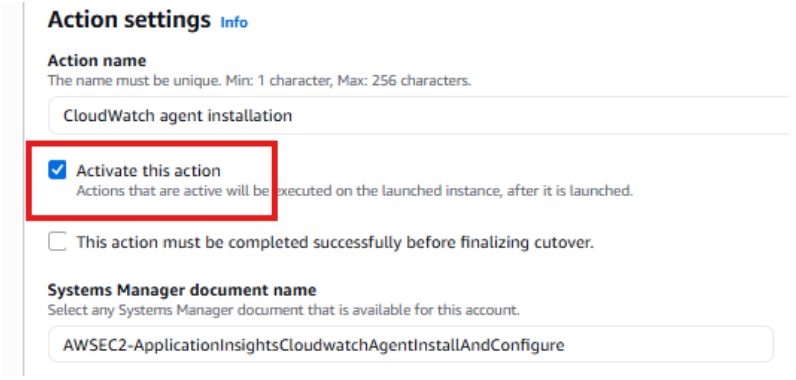
With this configuration we are going to install and configure AWS CloudWatch agent, a daemon that will publish specific metrics from the migrated instance. For this use case, we are going to configure the agent to push information about memory and swap utilization.

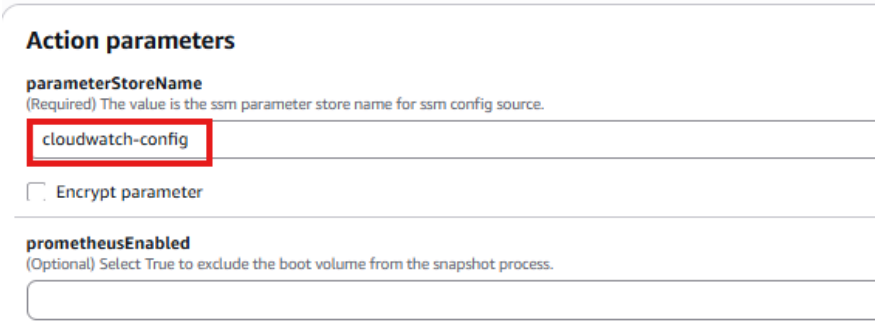


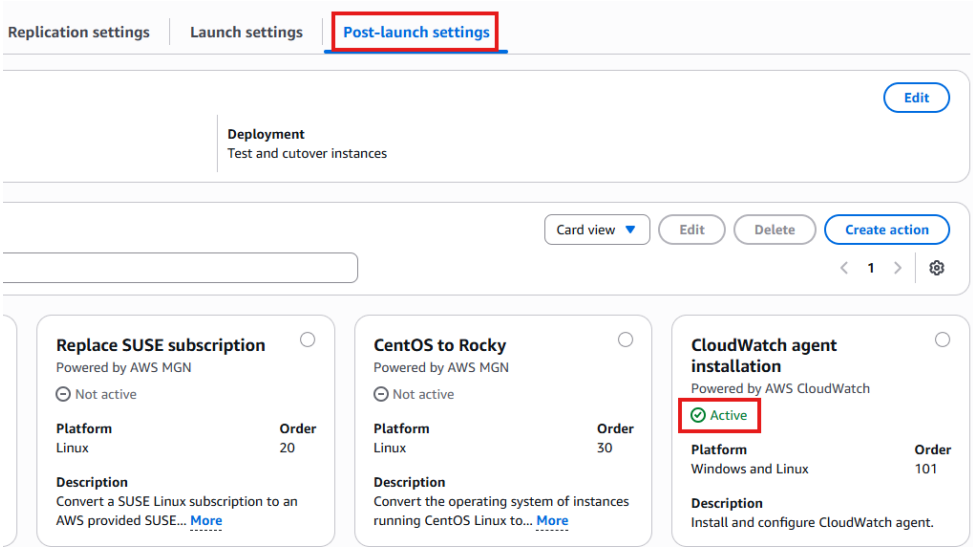




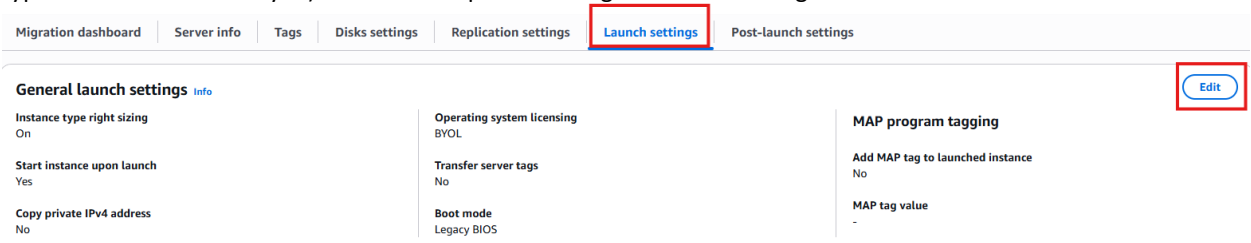








At Launch settings tab, you can configure details of how EC2 instance of replicated source server should be configured in AWS. For this use case, we will disable Instance type right sizing option (recommendation of instance type based on MGN analysis) and define expected configuration of the migrated EC2 instance.



ip-10-0-0-46

Launch Settings Info

EC2 Launch Template
lt-07e6d95081017f03c

Instance type right sizing Info
When Instance type right sizing is active, the instance type defined in the EC2 Launch template is ignored.

Off

Migration dashboard | Server info | Tags | Disks settings | Replication settings | Launch settings | Post-launch settings

General launch settings Info

Edit

Instance type right sizing
Off

Start instance upon launch
Yes

Copy private IPv4 address
No

Operating system licensing
BYOL

Transfer server tags
No

Boot mode
Legacy BIOS

MAP program tagging

Add MAP tag to launched instance
No

MAP tag value
-

EC2 Launch Template Info

Modify

Template ID
lt-07e6d95081017f03c

Instance type (ignored when right-sizing is active)
c5.large

Primary network interface
Description
-

Subnet

About modifying EC2 Launch Templates

Application Migration Service always uses the "Default" version of the EC2 Launch Template.

Every time you change an EC2 Launch Template, a new version is created. In order for Application Migration Service to use the new version you created, you must mark it as "Default". Learn more

Close | Modify

EC2 > Launch templates > Modify template (Create new version)

Recents | Quick Start

Don't include in launch template | Currently in use

Instance type Info | Get advice

Instance type

t3.micro — Free tier eligible

Family: t3 2 vCPU 1 GiB Memory Current generation: true On-Demand SUSE base pricing: 0.0104 USD per Hour
On-Demand Ubuntu Pro base pricing: 0.0139 USD per Hour On-Demand Windows base pricing: 0.0196 USD per Hour
On-Demand RHEL base pricing: 0.0392 USD per Hour On-Demand Linux base pricing: 0.0104 USD per Hour

Additional costs apply for AMIs with pre-installed software

Network settings Info

Subnet | Info

subnet-0ee3093e358fdae5d
VPC: vpc-0dba11cbb944fb462 Owner: 381244258876 Availability Zone: us-west-2a (usw2-az1) Zone type: Availability Zone
IP addresses available: 248 CIDR: 10.0.0.0/24)

cloud-app-vpc-public-a

When you specify a subnet, a network interface is automatically added to your template.

Availability Zone | Info

Don't include in launch template

Firewall (security groups) | Info

A security group is a set of firewall rules that control the traffic for your instance. Add rules to allow specific traffic to reach your instance.

- ○ Select existing security group
- ◉ Create security group

Security group name - *required*

MigratedWebServer-SG

This security group will be added to all network interfaces. The name can't be edited after the security group is created. Max length is 255 characters. V

Description - *required* | Info

Allow HTTP from anywhere

VPC | Info

vpc-0dba11cbb944fb462

Inbound Security Group Rules

No security group rules are currently included in this template. Add a new rule to include it in the launch template.

Add security group rule

Type	Protocol	Port range
HTTP	TCP	80

Source type	Source	Description - *optional*
Custom	Add CIDR, prefix list or security group 0.0.0.0/0	e.g. SSH for admin desktop

⚠ Rules with source of 0.0.0.0/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only.

Add security group rule

▶ Advanced network configuration

▼ Advanced network configuration

Network interface 1

Device index	Network interface	Description
0	New interface	

Subnet	Security groups	Auto-assign public IP
subnet-0ee3093e358fdae5d IP addresses available: 248	New security group	Enable

Resource tags Info

Key	Value	Resource types	
AWSApplicationMigrationServiceM	mgn.amazonaws.com	Instances, Volumes	Remove
AWSApplicationMigrationServiceS	s-5d85be895d0da5286	Instances, Volumes	Remove
Name	Webserver	Instances, Volumes	Remove

Add new tag

Advanced details Info

IAM instance profile | Info

ApplicationMigrationWorkshop-EC2InstanceProfile-iEcs1d9OmJ3m
arn:aws:iam::381244258876:instance-profile/ApplicationMigrationWorkshop-EC2InstanceProfile-iEcs1d9OmJ3m

Hostname type | Info

Don't include in launch template

Metadata response hop limit | Info

Don't include in launch template

Allow tags in metadata | Info

Don't include in launch template

User data - *optional* | Info

Upload a file with your user data or enter it in the field.

Choose file

```
apt remove -y unattended-upgrades
```

Summary

Software Image (AMI)
-

Virtual server type (instance type)
t3.micro

Firewall (security group)
New security group

Storage (volumes)
2 volume(s) - 24 GiB

Cancel | Create template version

EC2 > Launch templates > Modify template (Create new version)

Success
Successfully created created-and-used-by-application-migration-service-s-5d85be895d0da5286-251611-213544(lt-07e6d95081017f03c).

Actions log

Next Steps

Launch an instance

With On-Demand Instances, you pay for compute capacity by the second (for Linux, with a minimum of 60 seconds) or by the hour (for all other operating systems) with no long-term commitments or upfront payments. Launch an On-Demand Instance from your launch template.

Launch instance from this template

Create an Auto Scaling group from your template

Amazon EC2 Auto Scaling helps you maintain application availability and allows you to scale your Amazon EC2 capacity up or down automatically according to conditions you define. You can use Auto Scaling to help ensure that you are running your desired number of Amazon EC2 instances during demand spikes to maintain performance and decrease capacity during lulls to reduce costs.

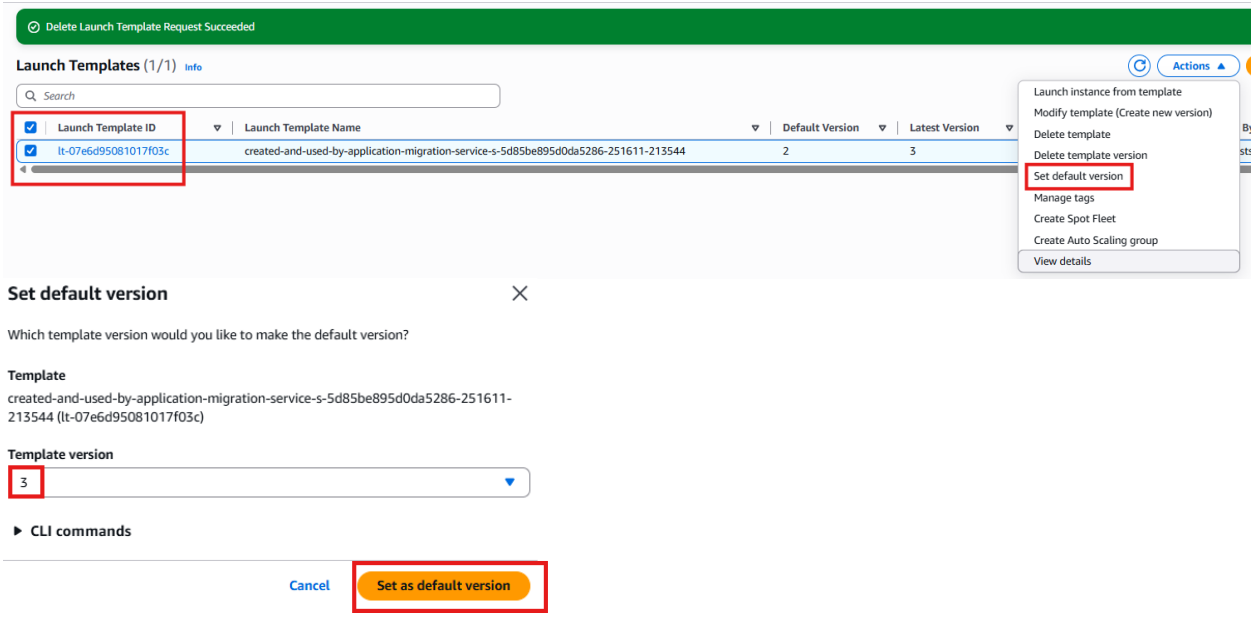
Create Auto Scaling group

Create Spot Fleet

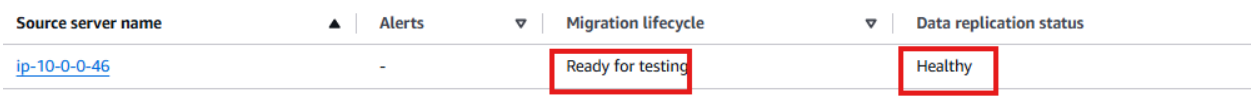
A Spot Instance is an unused EC2 instance that is available for less than the On-Demand price. Because Spot Instances enable you to request unused EC2 instances at steep discounts, you can lower your Amazon EC2 costs significantly. The hourly price for a Spot Instance (of each instance type in each Availability Zone) is set by Amazon EC2, and adjusted gradually based on the long-term supply of and demand for Spot Instances. Spot instances are well-suited for data-analysis, batch jobs, background processing, and optional tasks.

Create Spot Fleet

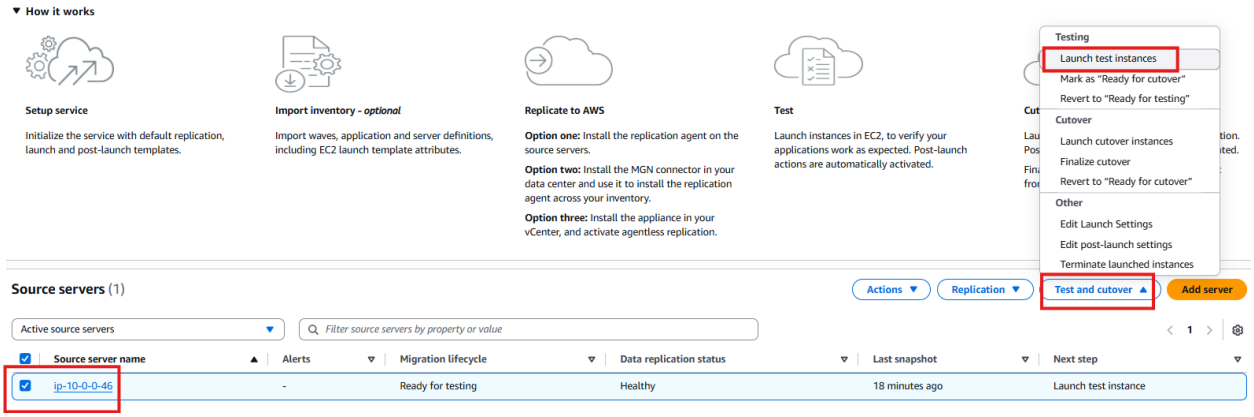
View launch templates



In the AWS Console, go back to Services > Migration & Transfer > Application Migration Service at Source servers menu, check if your source server is listed with Migration lifecycle as Ready for testing and Data replication status as Healthy.



In this activity of rehost migration of our source server, to validate if MGN is able to lift-and-shift it properly, we will have a sandbox instance and be able to confirm if the workload is fully functional after migration, including the boot volume conversion and the customized networking configuration.



Launch test instance for 1 server

You are about to launch EC2 instance for 1 server.

These instances will be launched according to the Launch Settings you have configured for them. Launched instances accrue EC2 charges as per your AWS account's rates. Learn more

▼ **The action will be applied to the following servers**

ip-10-0-0-46

Cancel **Launch**

Migration dashboard | Server info | Tags | Disks settings | Replication settings | Launch settings | Post-launch settings

Lifecycle Info

Not ready → Ready for testing → **Test in progress** → Ready for cutover → Cutover in progress → Cutover complete

Launch status
Waiting

Post-launch actions status
-

Last test
Job ID: mgnjob-5a44db4bf73acfc4c
Started: November 16, 2025 at 17:20 (UTC-6:00)

Cutover
-

Data replication status Info

Healthy

Replication progress
Initial replication finished

Replication type
Agent based

Total replicated storage
24 of 24 GiB

Lag
-

Backlog
-

Elapsed replication time
2 hr

Last seen
November 16, 2025 at 18:01 (UTC-6:00)

Replication start time
November 16, 2025 at 15:39 (UTC-6:00)

Replication instance ID
i-01b538d53bd54e38f

Post-launch actions (2) Info

Name	Execution status	Start time	End time	Details	Link
SSM Agent installed	Succeeded	November 16, 2025 at 17:34 (UTC-6:00)	November 16, 2025 at 17:34 (UTC-6:00)	-	Troubleshooting FAQ

▼ Launch history

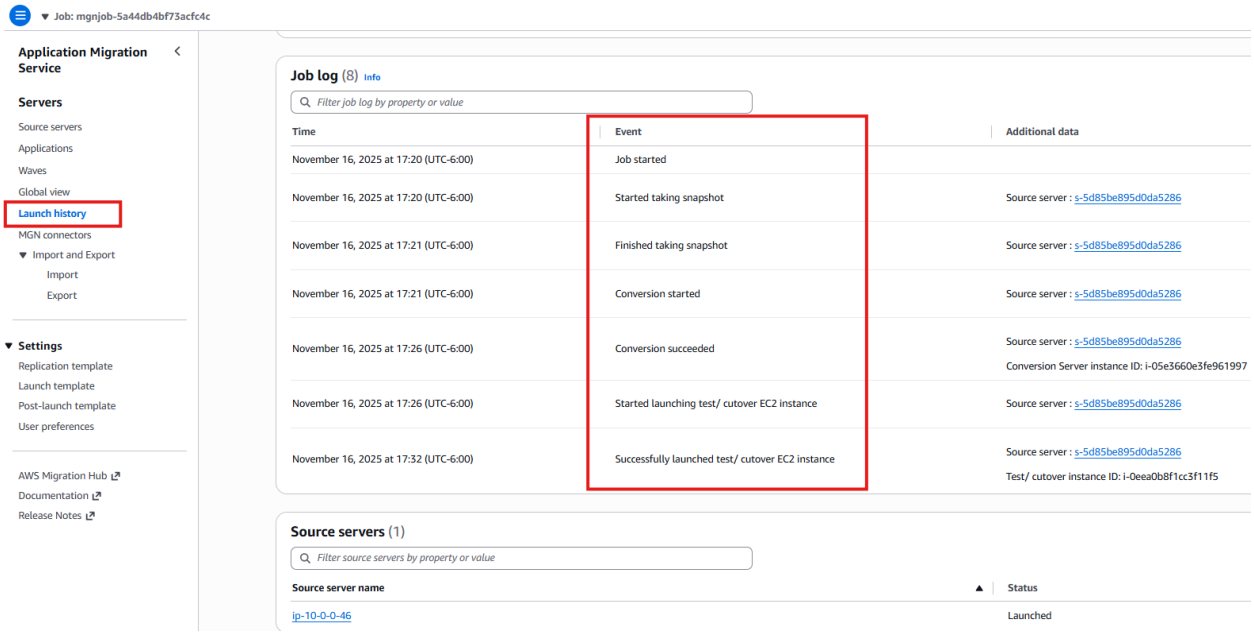
Application Migration Service

Servers
- Source servers
- Applications
- Waves
- Global view
- **Launch history**
- MGN connectors
- ▼ Import and Export

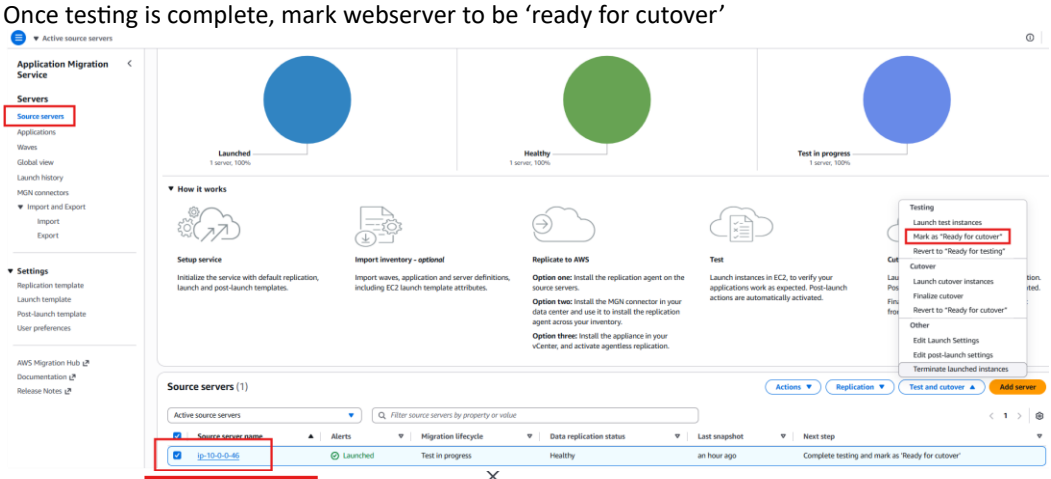
Launch history (4)

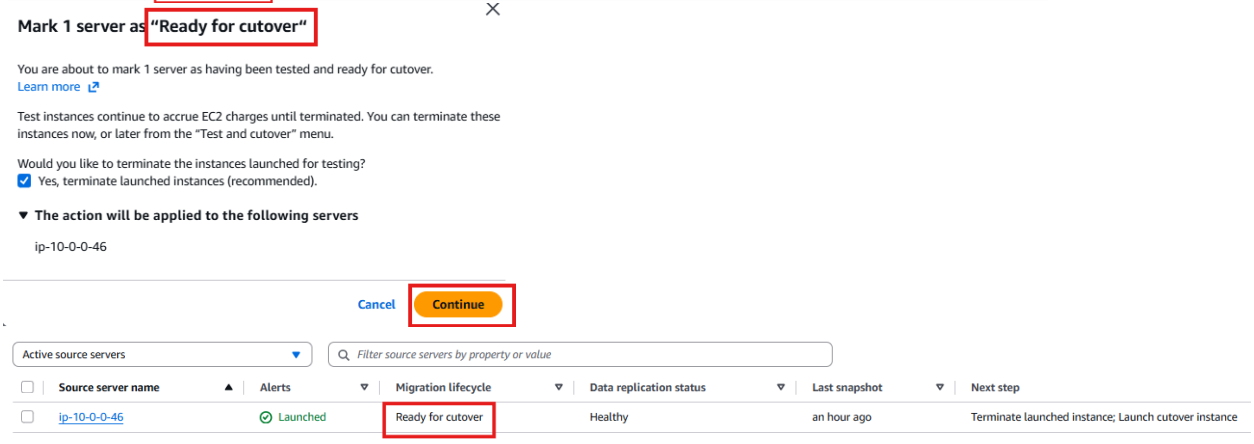
Filter launch history by property or value

Job ID	Job type	Initiated by	Status	Servers	Start time
mgnjob-5a44db4bf73acfc4c	Launch	Launch test instances	Completed	1	44 minutes ago
mgnjob-524685ffc01b36bd5	Launch	Launch cutover instances	Completed	1	a month ago
mgnjob-53ecc43d45d7e5f6e	Terminate	Terminate launched instances	Completed	1	a month ago
mgnjob-5b3a1590faab5e231	Launch	Launch test instances	Completed	1	a month ago

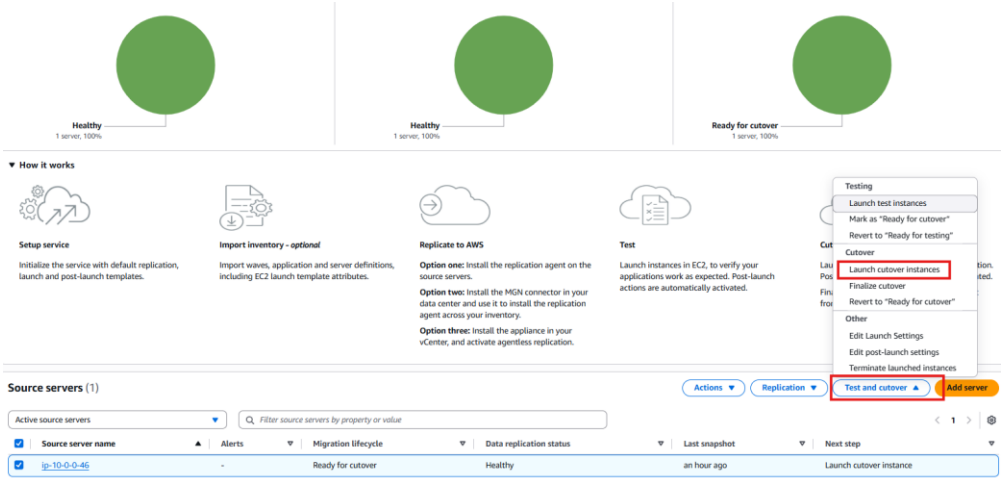


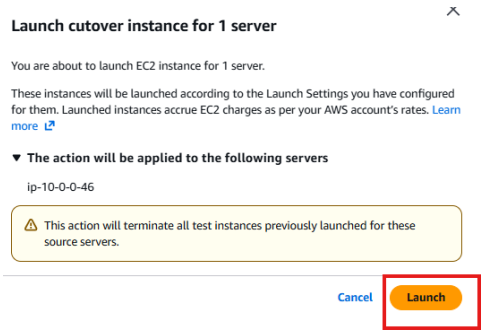
Once testing is complete, mark webserver to be 'ready for cutover'

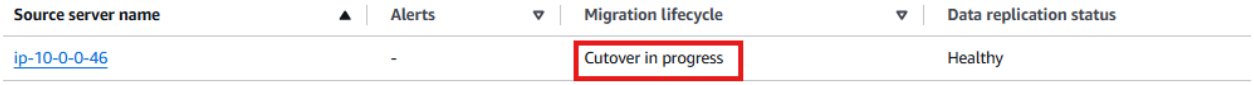


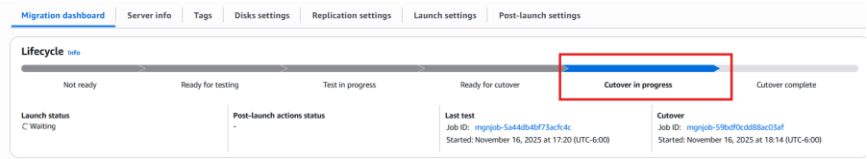


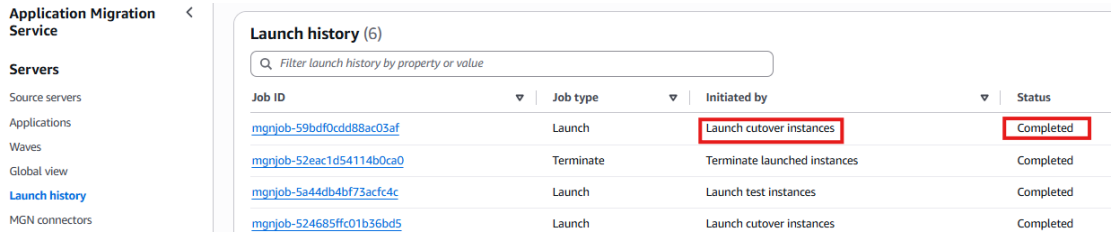
Now you are ready to launch the real migration by launching the cutover instance!

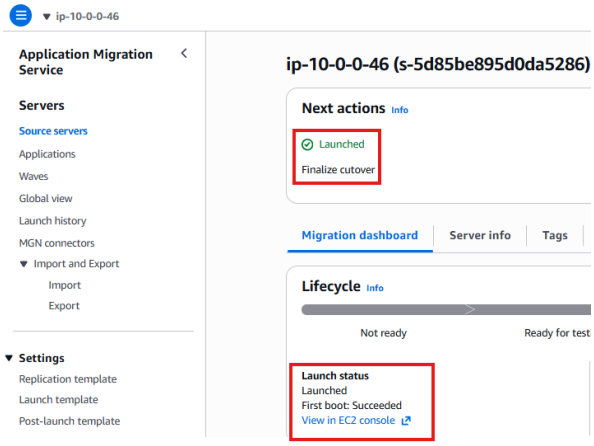

Healthy
1 server, 100%
Healthy
1 server, 100%
Ready for cutover
1 server, 100%
How it works
Setup service
Initialize the service with default replication, launch and post-launch templates.
Import inventory - optional
Import waves, application and server definitions, including EC2 launch template attributes.
Replicate to AWS
Option one: Install the replication agent on the source servers.
Option two: Install the MGN connector in your data center and use it to install the replication agent across your inventory.
Option three: Install the appliance in your vCenter, and activate agentless replication.
Test
Launch instances in EC2, to verify your applications work as expected. Post-launch actions are automatically activated.
Testing
Launch test instances
Mark as "Ready for cutover"
Revert to "Ready for testing"
Cutover
Launch cutover instances
Finalize cutover
Revert to "Ready for cutover"
Other
Edit Launch Settings
Edit post-launch settings
Terminate launched instances
Source servers (1)
Actions
Replication
Test and cutover
Add server
Active source servers
Filter source servers by property or value
Source server name
Alerts
Migration lifecycle
Data replication status
Last snapshot
Next step
ip-10-0-0-46
-
Ready for cutover
Healthy
an hour ago
Launch cutover instance


Launch cutover instance for 1 server
You are about to launch EC2 instance for 1 server.
These instances will be launched according to the Launch Settings you have configured for them. Launched instances accrue EC2 charges as per your AWS account's rates. Learn more
The action will be applied to the following servers
ip-10-0-0-46
This action will terminate all test instances previously launched for these source servers.
Cancel
Launch

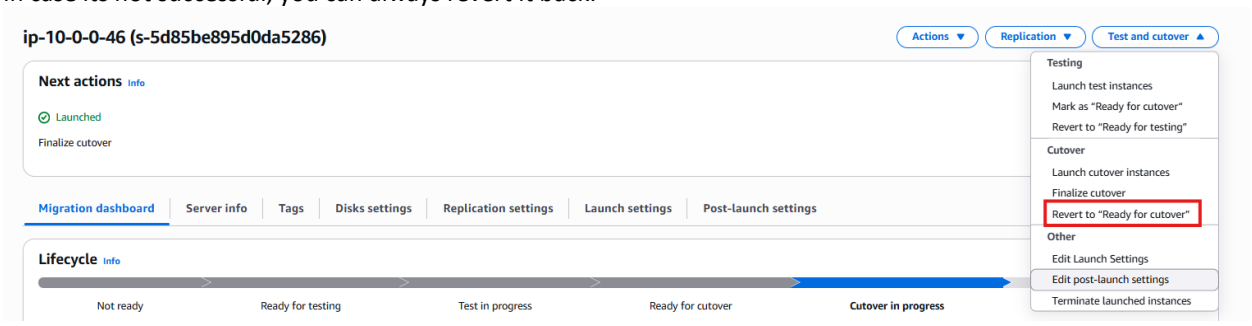

Source server name
Alerts
Migration lifecycle
Data replication status
ip-10-0-0-46
-
Cutover in progress
Healthy


Migration dashboard
Server info
Tags
Disks settings
Replication settings
Launch settings
Post-launch settings
Lifecycle Info
Not ready
Ready for testing
Test in progress
Ready for cutover
Cutover in progress
Cutover complete
Launch status
Waiting
Post-launch actions status
-
Last test
Job ID: mgnjob-5a44db4bf73acfc4c
Started: November 16, 2025 at 17:20 (UTC-6:00)
Cutover
Job ID: mgnjob-59bdf0cdd88ac03af
Started: November 16, 2025 at 18:14 (UTC-6:00)


Application Migration Service
Servers
Source servers
Applications
Waves
Global view
Launch history
MGN connectors
Launch history (6)
Filter launch history by property or value
Job ID
Job type
Initiated by
Status
mgnjob-59bdf0cdd88ac03af
Launch
Launch cutover instances
Completed
mgnjob-52eac1d54114b0ca0
Terminate
Terminate launched instances
Completed
mgnjob-5a44db4bf73acfc4c
Launch
Launch test instances
Completed
mgnjob-524685ffc01b36bd5
Launch
Launch cutover instances
Completed



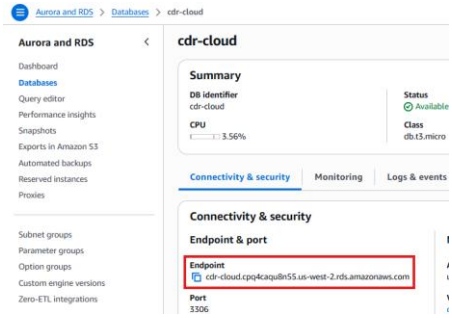
In case its not successful, you can always revert it back.



Configure the Webserver to access the target database

Open Webserver unix terminal session and run the following Then, let's modify the application configuration with the updated database information (configuration variables for **DB_USER**, **DB_PASSWORD**, **DB_HOST**). For that, run the script below replacing the <placeholders> with the correct value:

Get rds db endpoint-



sudo su -

*sed -i "s|^.*DB_USER.*$|define('DB_USER', 'admin');|" /var/www/html/wp-config.php*

*sed -i "s|^.*DB_PASSWORD.*$|define('DB_PASSWORD', 'Babla68??');|" /var/www/html/wp-config.php*

*** replace db_host value below from previous screen shot*

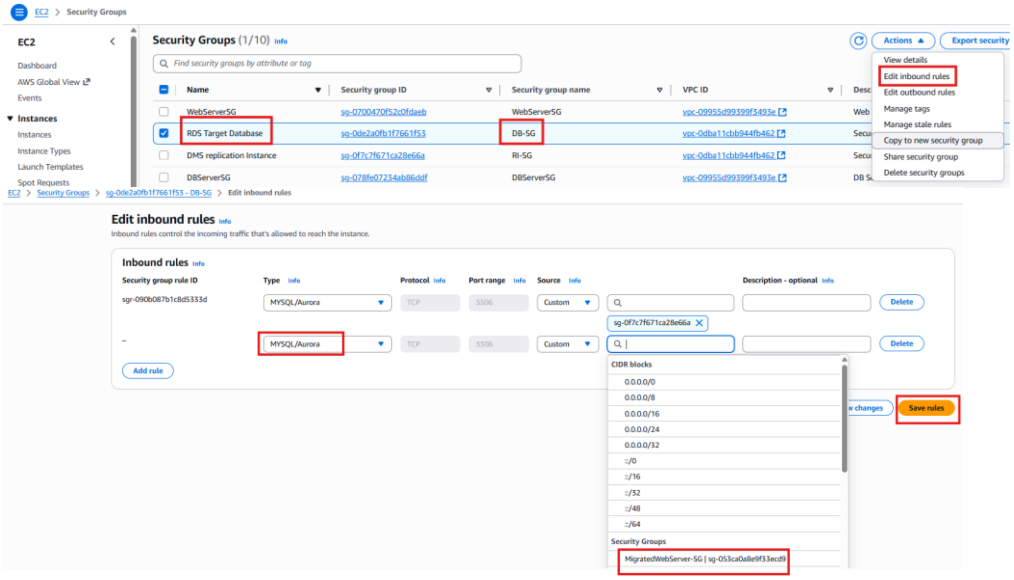
*sed -i "s|^.*DB_HOST.*$|define('DB_HOST', 'cdr-cloud.cpq4caqu8n55.us-west-2.rds.amazonaws.com');|" /var/www/html/wp-config.php*

echo "define('WP_SITEURL', 'http://$(curl -s http://169.254.169.254/latest/meta-data/public-hostname)');" >> /var/www/html/wp-config.php

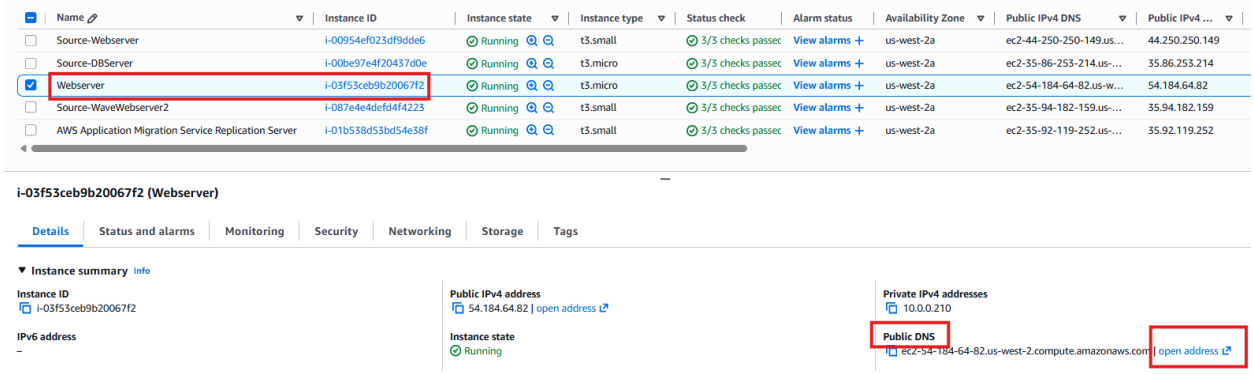
echo "define('WP_HOME', 'http://$(curl -s http://169.254.169.254/latest/meta-data/public-hostname)');" >> /var/www/html/wp-config.php

```
root@ip-10-0-0-210:~# sudo su -
root@ip-10-0-0-210:~# sed -i "s|^.*DB_USER.*$|define( 'DB_USER', 'admin' );|" /var/www/html/wp-config.php
root@ip-10-0-0-210:~# sed -i "s|^.*DB_PASSWORD.*$|define( 'DB_PASSWORD', 'Babla68?? );|" /var/www/html/wp-config.php
root@ip-10-0-0-210:~# sed -i "s|^.*DB_HOST.*$|define( 'DB_HOST', 'cdr-cloud.cpq4caqu8n55.us-west-2.rds.amazonaws.com' );|" /var/www/html/wp-config.php
root@ip-10-0-0-210:~# echo "define('WP_SITEURL', 'http://$(curl -s http://169.254.169.254/latest/meta-data/public-hostname)');" >> /var/www/html/wp-config.php
root@ip-10-0-0-210:~# echo "define('WP_HOME', 'http://$(curl -s http://169.254.169.254/latest/meta-data/public-hostname)');" >> /var/www/html/wp-config.php
```

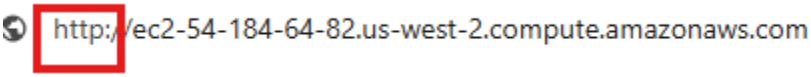
Now, we need to update the target database's Security Group to allow connectivity from the cutover EC2 instance. For that, in the AWS Console, access **Services > Compute > EC2**. Then click on **Security Group** menu in the left, select **DB-SG** entry, and from **Action** button select **Edit inbound rules**:



Verify the site I sup and running from target web server

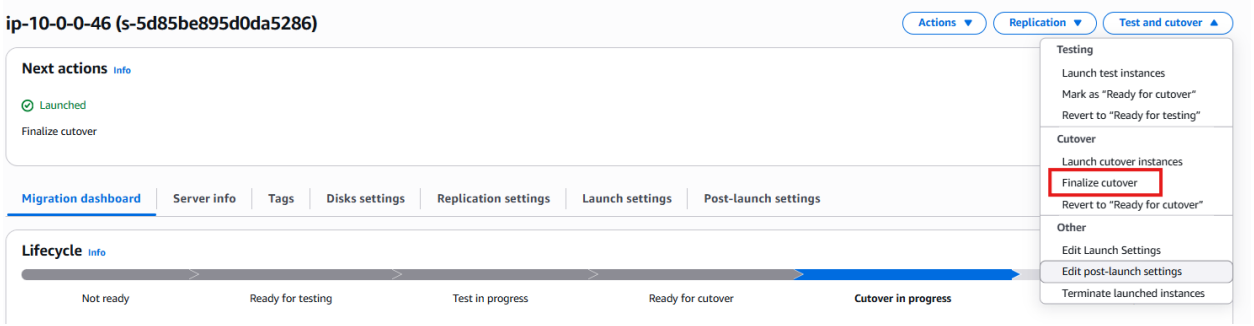


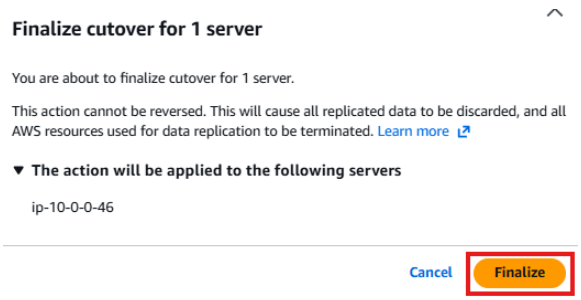
Make sure to manually change that to http after it opens in browser

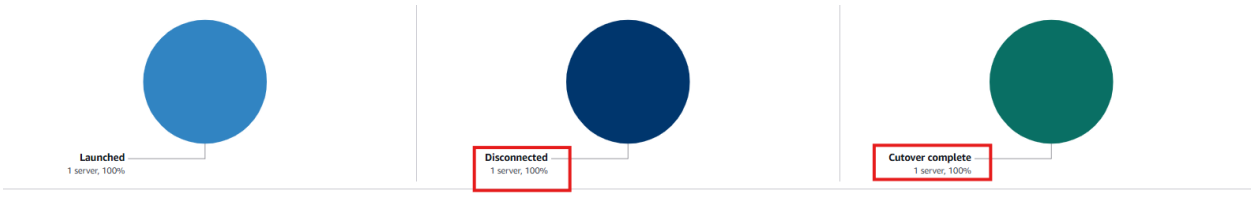


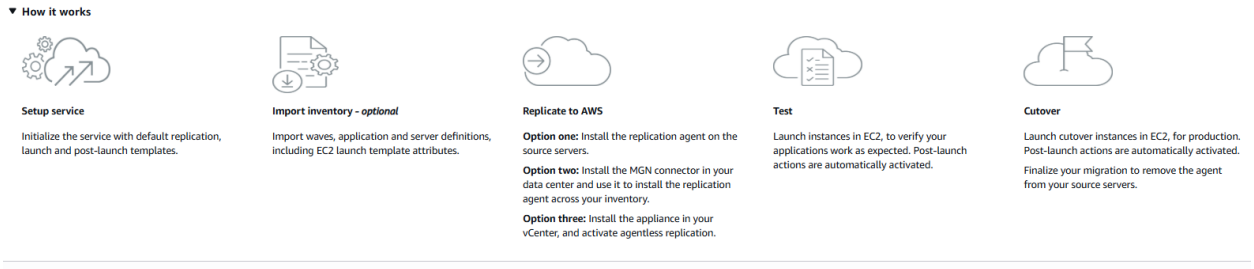
That should bring you to your site.

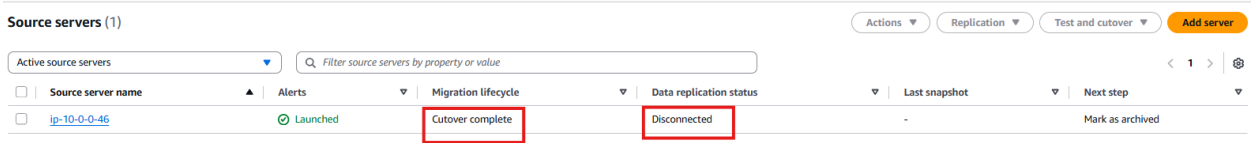
Finalize the cutover on MGN











Work completed:

1. Initialized and configured AWS Application Migration Service (MGN)
2. Deployed MGN Agent to the source application server
3. Customized the EC2 Launch Template to fine tune how the rehosted server would be configured in the target AWS environment
4. Optionally configured a post-launch action
5. Launched test instance with MGN to validate the rehost process was functional
6. Launched the cutover instance, and reconfigured the application to the new target environment
7. Finalized the cutover process, and executed housekeeping on AWS Application Migration Service (MGN) inventory

CLEANUP

Follow this, https://catalog.us-east-1.prod.workshops.aws/workshops/c6bdf8dc-d2b2-4dbd-b673-90836e954745/en-US/08-cleanup

At the end run the following from cloudshell terminal-

wget https://ws-assets-prod-iad-r-pdx-f3b3f9f1a7d6a3d0.s3.us-west-2.amazonaws.com/c6bdf8dc-d2b2-4dbd-b673-90836e954745/scripts/cleanupads.py

python cleanupads.py