

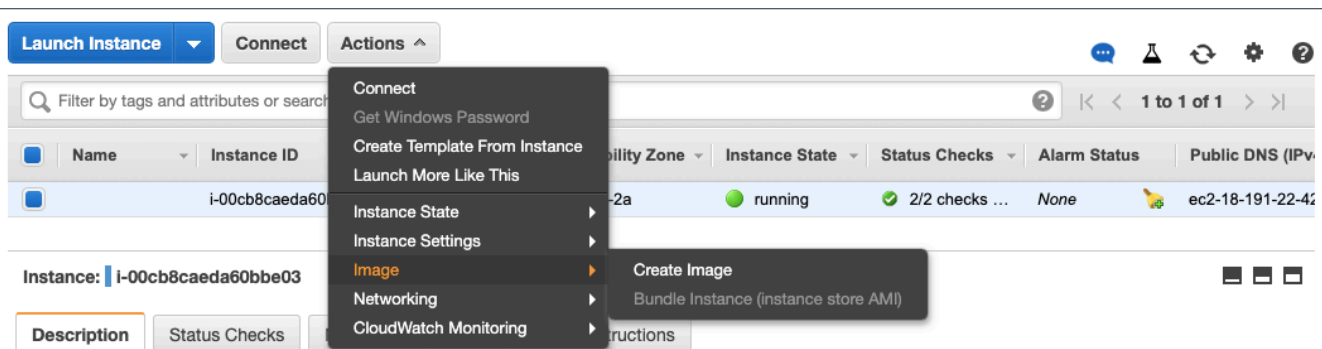
Server Tuning & Monitoring Commands

Steps involved in Server Tuning:

1. Take AMI Backup in the AWS EC2 console
2. Apache Tuning
3. Take DB Backup
4. Check the tables storage engine type and convert all MYISAM table to INNODB
5. Take a DB backup again after the conversion
6. MYSQL Tuning
7. Instance upgrade/Downgrade
8. Important commands to check the changes

1, Take AMI Backup in the AWS EC2 console:

- Go to EC2 console in the AWS
- Click Actions button
- Click Image and then click create image



- Give the image name and image description, select No reboot and click create image
- In the AMI section check the status, once status shows available proceed to the apache tuning until wait on

Create Image

Instance ID ⓘ

i-00cb8caeda60bbe03

Image name ⓘ

11Oct21

Image description ⓘ

11Oct21

No reboot ⓘ

☒

Instance Volumes

Volume Type ⓘ	Device ⓘ	Snapshot ⓘ	Size (GiB) ⓘ	Volume Type ⓘ	IOPS ⓘ	Throughput (MB/s) ⓘ	Delete on Termination ⓘ	Encrypted ⓘ	
Root	/dev/sda1	snap-000e8f72d5310410b	20	General Purpose S3	100 / 3000	N/A	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Not Encrypted

Add New Volume

Total size of EBS Volumes: 20 GiB

When you create an EBS image, an EBS snapshot will also be created for each of the above volumes.

Cancel

Create Image

The screenshot shows the AWS Management Console interface for an Amazon Machine Image (AMI). The left sidebar contains navigation links for Tags, Limits, Instances, Instance Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, Capacity Reservations, Images, and AMIs. The main content area displays the details for the AMI '11Oct21' with ID 'ami-0311cab57818a7541'. The AMI is owned by '205042197256/1...' and is in an 'available' status. The creation date is 'October 11, 2021 at 3:53:45 PM UTC+5:30'. The architecture is 'x86_64'. The platform details are 'Linux/UNIX' and the usage operation is 'RunInstances'.

Name	AMI Name	AMI ID	Source	Owner	Visibility	Status	Creation Date
11Oct21	ami-0311cab57818a7541	ami-0311cab57818a7541	205042197256/1...	205042197256	Private	available	October 11, 2021 a

Image: ami-0311cab57818a7541

Property	Value
AMI ID	ami-0311cab57818a7541
Owner	205042197256
Status	available
Creation date	October 11, 2021 at 3:53:45 PM UTC+5:30
Architecture	x86_64
AMI Name	11Oct21
Source	205042197256/11Oct21
State Reason	-
Platform details	Linux/UNIX
Usage operation	RunInstances

2, Apache tuning:

Tuning Calculation:

Max Request Workers: $\text{RAM} * 150$

Server Limit: $\text{RAM} * 256$

Here 150 and 256 is the default value in Max Request and server limit in Apache.

- Login to the ssh of the server
- Go to `cd /etc/apache2/conf/`
- Then `ls`
- Take backup of `httpd.conf` (`cp httpd.conf httpd.conf.bak`)
- Change the Max Request Workers 2400 & Server Limit 4096
- Then restart apache server (`sudo systemctl restart httpd`)

```

GNU nano 2.3.1      File: /etc/apache2/conf/httpd.conf      Modified

</Directory>

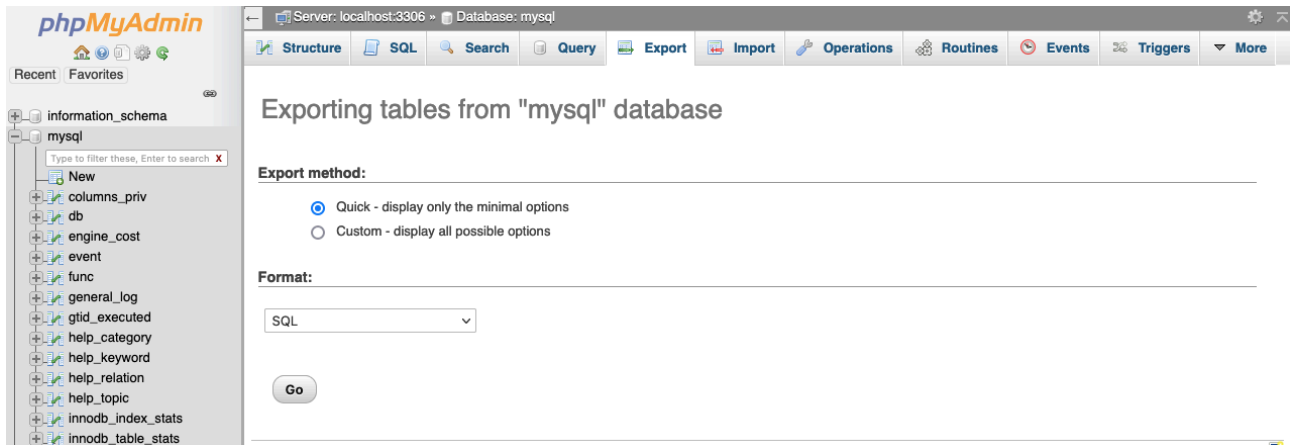
StartServers 5
<IfModule prefork.c>
    MinSpareServers 5
    MaxSpareServers 10
</IfModule>

ServerLimit 4096
MaxRequestWorkers 2400
MaxConnectionsPerChild 10000
KeepAlive On
KeepAliveTimeout 5
MaxKeepAliveRequests 100
Timeout 300

```

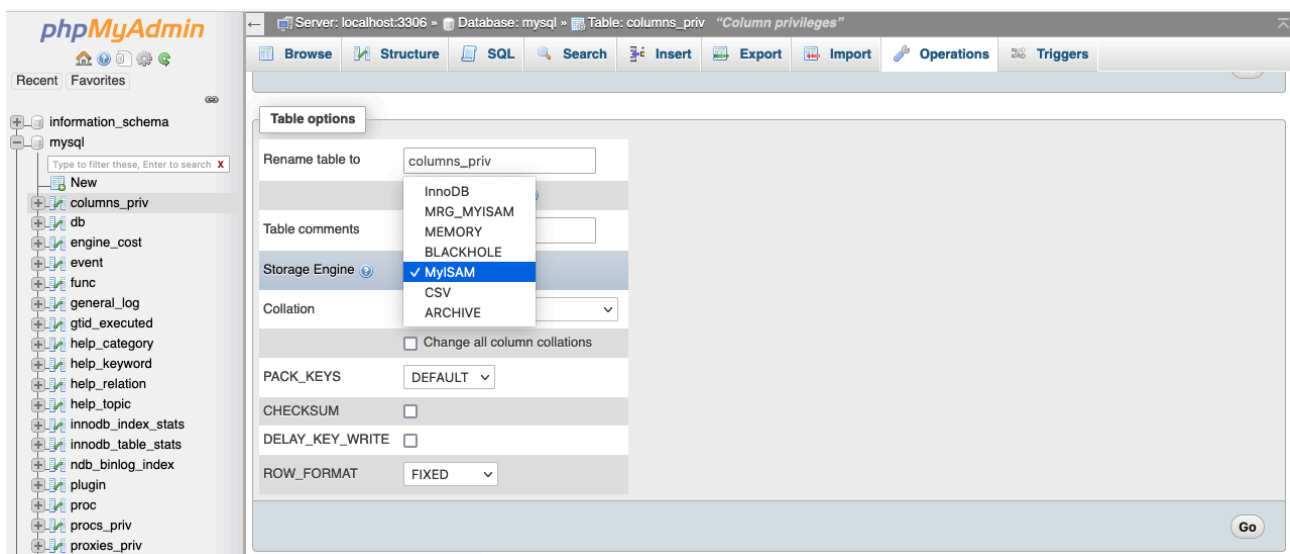
3, Take DB Backup:

- Go to PHPMYADMIN of the server
- Select the DB
- Click Export and click go
- Now the DB will download to your system



4, Check the DB storage engine & conversion:

- Select the DB
- Now the under the the type you can find the storage engine type
- Click the MYISAM table here in the below snapshot I have selected columns_priv table
- Click the operations button and then under the table options change the storage engine to INNODB and click go
- Now the DB has changed to INNODB



5, Repeat the 3rd step

6, MYSQL Tuning

Tuning Calculation:

$\text{innodb_buffer_pool_size} = \text{RAM} * 1024\text{M} * 50\%$

$\text{innodb_log_buffer_size} = \text{RAM} * 1024\text{M} * 10\%$

In the MYSQL tuning file change only the above mentioned values other values remains the same.

Here in the below configuration file the value is calculated for 16Gib RAM instance type is m4.xlarge

- Stop MySQL server (sudo systemctl stop mysqld)
- Go to cd /var/lib/mysql
- ls -ltrh (Check the mysql.sock)
- mv ib_logfile0 /tmp/
- mv ib_logfile1 /tmp/
- Edit the mysql config file (nano /etc/my.cnf) Add the ***MYSQL Tuning Values given below under instance upgrade/downgrade***

```
[root@18-191-22-42 centos]# cd /var/lib/mysql
[root@18-191-22-42 mysql]# ls -ltrh
total 121M
-rw-r-----. 1 mysql mysql 56 Oct 11 07:52 auto.cnf
-rw-----. 1 mysql mysql 1.7K Oct 11 07:52 ca-key.pem
-rw-r--r--. 1 mysql mysql 1.1K Oct 11 07:52 ca.pem
-rw-----. 1 mysql mysql 1.7K Oct 11 07:52 server-key.pem
-rw-r--r--. 1 mysql mysql 1.1K Oct 11 07:52 server-cert.pem
-rw-----. 1 mysql mysql 1.7K Oct 11 07:52 client-key.pem
-rw-r--r--. 1 mysql mysql 1.1K Oct 11 07:52 client-cert.pem
-rw-r--r--. 1 mysql mysql 452 Oct 11 07:52 public_key.pem
-rw-----. 1 mysql mysql 1.7K Oct 11 07:52 private_key.pem
drwxr-x---. 2 mysql mysql 8.0K Oct 11 07:52 sys
drwxr-x---. 2 mysql mysql 8.0K Oct 11 07:52 performance_schema
drwxr-x---. 2 mysql mysql 4.0K Oct 11 07:52 mysql
-rw-----. 1 root root 6 Oct 11 07:52 mysql_upgrade_info
-rw-r-----. 1 mysql mysql 362 Oct 11 10:29 ib_buffer_pool
-rw-r-----. 1 mysql mysql 48M Oct 11 10:30 ib_logfile1
-rw-----. 1 mysql mysql 5 Oct 11 10:30 mysql.sock.lock
srwxrwxrwx. 1 mysql mysql 0 Oct 11 10:30 mysql.sock
-rw-r-----. 1 mysql mysql 12M Oct 11 10:30 ibdata1
-rw-r-----. 1 mysql mysql 48M Oct 11 10:30 ib_logfile0
-rw-r-----. 1 mysql mysql 12M Oct 11 10:50 ibtmp1
[root@18-191-22-42 mysql]#
```

```
# Remove leading # to set options mainly useful for reporting servers.
# The server defaults are faster for transactions and fast SELECTs.
# Adjust sizes as needed experiment to find the optimal values.
# join_buffer_size = 128M
# sort_buffer_size = 2M
# read_rnd_buffer_size = 2M
datadir=/var/lib/mysql
socket=/var/lib/mysql/mysql.sock

# Disabling symbolic-links is recommended to prevent assorted security risks
symbolic-links=0

log-error=/var/log/mysqld.log
pid-file=/var/run/mysqld/mysqld.pid
innodb_file_per_table=1
#max_allowed_packet=268435456
open_files_limit=40000
sql_mode=NO_ENGINE_SUBSTITUTION

#GLOBAL BUFFERS

connect_timeout=300

#GLOBAL BUFFERS
query_cache_limit=20M
query_cache_size = 1024M

innodb_data_file_path = ibdata1:1024M:autoextend
innodb_buffer_pool_size =8200M

innodb_log_file_size =1024M
innodb_log_buffer_size =1500M
innodb_flush_log_at_trx_commit = 2
innodb_lock_wait_timeout = 50
innodb_thread_concurrency=12
innodb_file_per_table=1
innodb_log_files_in_group=2

# Session Buffers

sort_buffer_size =16M
join_buffer_size=16M
net_buffer_length = 1M
read_buffer_size =16M
```

G Get Help
X Exit

O WriteOut
J Justify

R Read File
W Where Is

Y Prev Page
V Next Page

K Cut Text
U UnCut Text

C Cur Pos
T To Spell

7, Instance Upgrade/Downgrade

- Go to EC2 console in the AWS
- Click Actions button and then then instance state
- Click Stop
- Once the server is stopped click Actions button again
- Click instance settings and then change instance type and then choose the instance type as per your recommendation

The screenshot shows the AWS Management Console interface for an EC2 instance. The instance is named 'i-00cb8caeda60bbe03' and is in a 'stopped' state. The 'Actions' menu is open, showing options like 'Connect', 'Get Windows Password', 'Create Template From Instance', 'Launch More Like This', 'Instance State', 'Instance Settings', 'Image', 'Networking', and 'CloudWatch Monitoring'. The 'Instance Settings' submenu is also open, showing options like 'Add/Edit Tags', 'Attach to Auto Scaling Group', 'Attach/Replace IAM Role', 'Change Instance Type' (highlighted), 'Change Termination Protection', 'View/Change User Data', 'Change Shutdown Behavior', 'Change T2/T3 Unlimited', 'Change Nitro Enclaves', 'Get System Log', 'Get Instance Screenshot', 'Modify Instance Placement', and 'Modify Capacity Reservation Settings'.

Name	Instance ID	Availability Zone	Instance State	Status Checks	Alarm Status	Public DNS (IPv4)
	i-00cb8caeda60bbe03	us-east-2a	stopped	None		ec2-18-191-22-42

Instance: **i-00cb8caeda60bbe03**

Description	Status Checks
<p>Instance ID: i-00cb8caeda60bbe03</p> <p>Instance state: stopped</p> <p>Instance type: m4.xlarge</p> <p>Finding: Opt-in to AWS Compute Optimizer for recommendations. Learn more</p> <p>Private DNS: ip-172-31-15-119.us-east-2.compute.internal</p> <p>Private IPs: 172.31.15.119</p> <p>Secondary private IPs: vpc-dbdbb1b0</p> <p>VPC ID: vpc-dbdbb1b0</p>	<p>Scheduled events: -</p> <p>AMI ID: packer-2424047-8196214-99abd91c-0000-0000-0000-000000000000</p>

- Here in the example I'm upgrading the server to m4.xlarge
- Then Start the instance in the Actions button of the instance state

MYSQL Tuning Values:

#GLOBAL BUFFERS

connect_timeout=300

#GLOBAL BUFFERS

query_cache_limit=20M

query_cache_size = 1024M

#innodb_data_file_path = ibdata1:1024M:autoextend

innodb_buffer_pool_size =8192M

innodb_log_file_size =1024M

innodb_log_buffer_size =1638M

innodb_flush_log_at_trx_commit = 2

innodb_lock_wait_timeout = 50

innodb_thread_concurrency=12

innodb_file_per_table=1

innodb_log_files_in_group=2

Session Buffers

sort_buffer_size =16M

join_buffer_size=16M

net_buffer_length = 1M

read_buffer_size =16M

read_rnd_buffer_size =64M

myisam_sort_buffer_size = 128M

General Variables

back_log = 200

long_query_time=10

innodb_flush_method=O_DIRECT

thread_cache_size =256

delayed_queue_size = 1500

max_connections = 6500

max_connect_errors = 99999

max_allowed_packet=1024M

8, Important commands to check the changes and server monitoring

- Login to the ssh server
- mysql
- show variables like '%innodb_buffer%';

Steps to check in mysql server during server slow:

- show processlist;
- show status like '%max%';

Steps to check in Apache server during server slow:

- top (Shows cpu usage and task running in the server)
- w (load check)
- Free -m (To check the memory used)

Steps to check storage size:

- df - Th (Checks the entire storage size of the server)
- du -sh foldername (To check for a specific folder size)