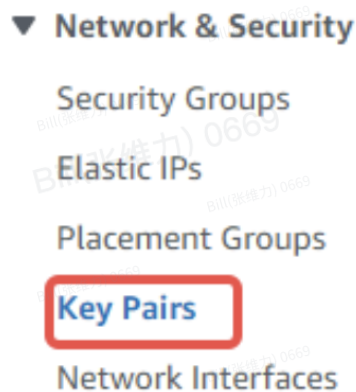


# GEARS QUICK START

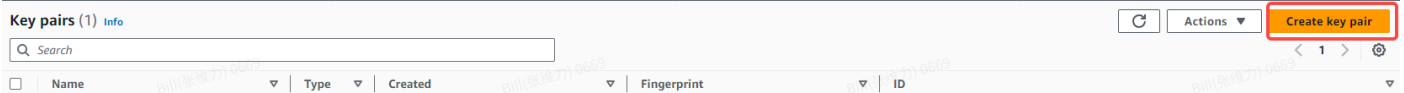
## Step 0: Log in to the EC2 console

## Step 1: Create a key pair (optional)

1. On the left, select "Key Pair" under "Network and Security".



2. Click on the top right corner to create a key pair



3. Enter a name, create a key pair, and select the private key file format based on your connection method

[EC2](#) > [Key pairs](#) > Create key pair

## Create key pair [Info](#)

**Key pair**  
A key pair, consisting of a private key and a public key, is a set of security credentials that you use to prove your identity when connecting to an instance.

**Name**  
  
The name can include up to 255 ASCII characters. It can't include leading or trailing spaces.

**Key pair type** [Info](#)  
☒ RSA ☐ ED25519

**Private key file format**  
☒ .pem  
For use with OpenSSH  
☐ .ppk  
For use with PuTTY

**Tags - optional**  
No tags associated with the resource.

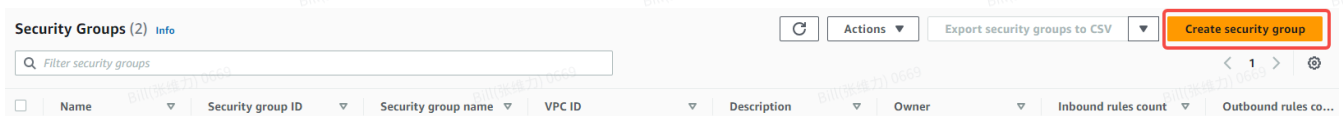
You can add up to 50 more tags.

## Step 2: Create a security group

1. On the left, select "Security Groups" under "Network & Security".



2. Click on "Create Security Group" in the upper right corner



3. Ensure that the following ports are open for entry rules: 22, 80, 8070, 8085, 8091.

| Type info  | Protocol info | Port range info | Source info | Description - optional info |                    |
|------------|---------------|-----------------|-------------|-----------------------------|--------------------|
| HTTP       | TCP           | 80              | Custom      | Q                           | 0.0.0.0/0 X Delete |
| Custom TCP | TCP           | 8091            | Custom      | Q                           | 0.0.0.0/0 X Delete |
| SSH        | TCP           | 22              | Custom      | Q                           | 0.0.0.0/0 X Delete |
| Custom TCP | TCP           | 8085            | Custom      | Q                           | 0.0.0.0/0 X Delete |
| Custom TCP | TCP           | 8070            | Custom      | Q                           | 0.0.0.0/0 X Delete |

Cancel Preview changes Save rules

### Step 3: Launch an Instance from AMI

1. In the console, open the detailed page of the GEARS AMI and start an instance using the button in the upper right corner.

EC2 > AMIs > ami-0cf9ce16bcf91d9c0

Image summary for ami-0cf9ce16bcf91d9c0

AMI ID: ami-0cf9ce16bcf91d9c0 Image type: machine Platform details: Linux/UNIX Root device type: EBS

AMI name: Owner account ID: Architecture: Usage operation:

EC2 Image Builder Actions Launch instance from AMI

2. Enter a name for your server (e.g., gears-xxx).

Name and tags Info

Name

e.g. My Web Server Add additional tags

3. Select an instance type with a configuration no lower than t2.xlarge.

Instance type Info

Instance type

t2.xlarge

Family: t2 4 vCPU 16 GiB Memory Current generation: true

On-Demand Linux base pricing: 0.1856 USD per Hour

On-Demand RHEL base pricing: 0.2456 USD per Hour

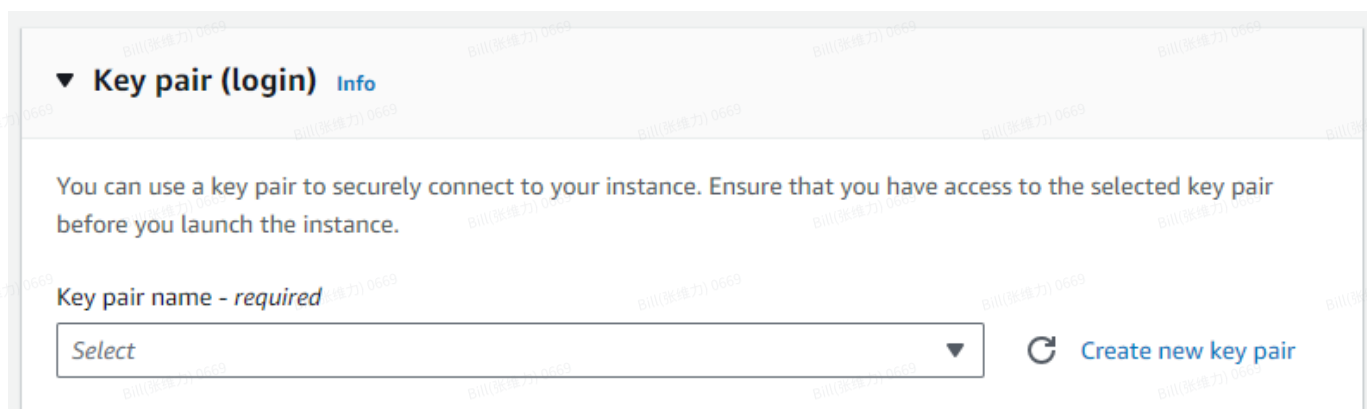
On-Demand Windows base pricing: 0.2266 USD per Hour

On-Demand SUSE base pricing: 0.2856 USD per Hour

Additional costs apply for AMIs with pre-installed software

All generations Compare instance types

#### 4. Select your key pair to log in to the server terminal



**▼ Key pair (login) Info**

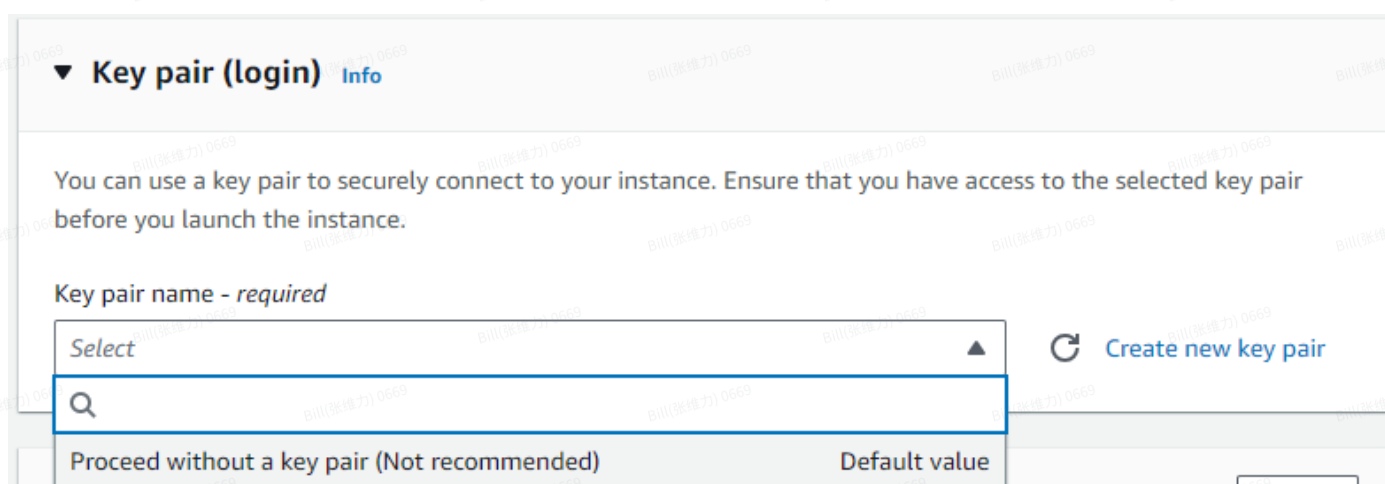
You can use a key pair to securely connect to your instance. Ensure that you have access to the selected key pair before you launch the instance.

Key pair name - *required*

Select ▼

↻ Create new key pair

If you don't want to use a key pair, you can choose to 'Proceed without a key pair' (not recommended!!)



**▼ Key pair (login) Info**

You can use a key pair to securely connect to your instance. Ensure that you have access to the selected key pair before you launch the instance.

Key pair name - *required*

Select ▲

Q

Proceed without a key pair (Not recommended) Default value

↻ Create new key pair

#### 5. In Network Settings, select the security group you just created.

▼ Network settings Info

Edit

Network Info

vpc-

Subnet Info

No preference (Default subnet in any availability zone)

Auto-assign public IP Info

Enable

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.

☐ Create security group

☒ Select existing security group

Common security groups Info

Select security groups

gears-sg sg- VPC: vpc-

×

Compare security group rules

Security groups that you add or remove here will be added to or removed from all your network interfaces.

6. For other configurations, you can use defaults unless you have specific requirements. Click "Launch Instance."

▼ Summary

Number of instances [Info](#)

1

Software Image (AMI)

gears

am

Virtual server type (instance type)

t2.xlarge

Firewall (security group)

gears-sg

Storage (volumes)

1 volume(s) - 30 GiB

**Free tier:** In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is unavailable) instance usage on free tier AMIs per month, 30 GiB of EBS storage, 2 million IOs, 1 GB of snapshots, and 100 GB of bandwidth to the internet.

Cancel **Launch instance**

[Review commands](#)

## Step 4: Connect to the Instance

1. After the instance starts, wait for approximately 5-10 seconds to ensure it is fully up and running.
2. Open a terminal and use SSH to connect to your instance using your key pair:

```
1 ssh -i "<path_to_key>/<key_file>" ec2-user@<public_IP_of_instance>
2 # e.g., ssh -i "D:/path/to/your/key/xxx.pem" ec2-user@111.222.111.222
```

## Step 5: Configure the Database

1. Start the MySQL service in a location that is accessible over the public network (e.g., RDS, cloud server).
2. It is recommended to use MySQL version 5.7.42, which should not be higher than 8.0. Incompatibility may occur, resulting in service connectivity issues

## Step 6: Initialize the Gears System

1. Navigate to the /home/ec2-user folder (or ~ if you are using the ec2-user login):

```
1 cd /home/ec2-user
```

2. Run the following command to configure the Gears system:

```
1 ./init.sh
```

3. Enter the database's address, port, username, password.
4. If your database service was just created and it's the first time using it, follow the prompts to enter 'y' for automatic database import.

## Step 7: Configure Apollo

1. After running init.sh, you will be prompted with the Apollo address.

(The Apollo address is usually the public IP address followed by port 8070, e.g., `http://<public_IP_of_instance>:8070`)

2. Access this address via a web browser. The initial Apollo admin credentials are:

- Username: apollo
- Password: admin

3. Select "Create Project" in the upper right corner.



4. For the initial creation, you can follow these settings, ensuring that the Appid is set to "gears." You can make other changes as needed.

Create Project

\* Department: 样例部门1(TEST1)

\* App Id: gears  
(Application's unique identifiers)

\* App Name: gears  
(Suggested format xx-yy-zz e.g. apollo-server)

\* App Owner: apollo | apollo | apollo@acme.com

App Administrators: apollo | apollo | apollo@acme.com  
(The application owner has project administrator permission by default. Project administrators can create namespace, cluster, and assign user permissions)

Submit

- After clicking "Submit", enter the application and click on the "Text" tab on the upper-left corner to access the batch configuration import page.

Environments: DEV

Project Info: App Id: gears, App Name: gears, Department: 样例部门1 (TEST1), App Owner: apollo(apollo), Email: apollo@acme.com

application

Table Text Change History Instance List

Release Status Key Type Value Comment Last Modifier Last Modified Time Operation

Tips: This namespace has never been released. Apollo client will not be able to fetch the configuration and will record 404 log information. Please release it in time.

- Click the "Edit" button in the upper right corner.

Environments: DEV

Project Info: App Id: gears, App Name: gears, Department: 样例部门1 (TEST1), App Owner: apollo(apollo)

application

Table Text Change History Instance List

Edit

- Enter the following configuration, modifying the first five lines with the database's address and information, and click "Confirm" in the upper right corner.

Private properties

application

Table Text Change History Instance List

27 mq.topic.sendmail = gears.topic.sendmail  
28 mq.queue.sendmail = gears.queue.sendmail  
29 spring.redis.host = host.docker.internal  
30 spring.redis.password = gears@123  
31 spring.redis.port = 6379  
32 spring.redis.database = 0  
33 spring.redis.timeout = 60s  
34 spring.redis.jedis.pool.max-idle = 0  
35 spring.redis.jedis.pool.min-idle = 0  
36 spring.redis.jedis.pool.max-wait = -1s  
37 spring.redis.jedis.pool.max-active = -1  
38 gears.jenkins.password = 11111111  
39 gears.jira-service.api = https://gears-service-dev.XXXXX.com/gears-jira/api  
40 gears.jira-service.token = gears\_jira\_007  
41 mq.queue.deploy = gears.queue.deploydev  
42 mq.queue.log = gears.queue.logdev  
43 X-TOKEN = 1a2p6F3uLsu0R77laoRghv3q  
44 steampipe-api-gateway.base\_url = http://steampipe-api-gateway.steampipe-dev.xxx.xxx:3000  
45 steampipe-api-gateway.client\_id = gears  
46 steampipe-api-gateway.client\_secret = 11111111

Confirm

```
1 gears.db.host = <database_host>
2 gears.db.port = <database_port>
```

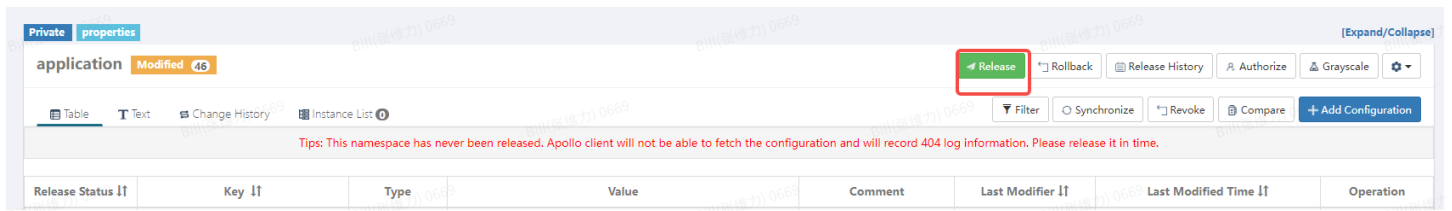


```

3 gears.db.username = <database_username>
4 gears.db.password = <database_password>
5 gears.db.database = gears
6 server.port = ${SERVER_PORT:8090}
7 spring.main.allow-bean-definition-overriding = true
8 mybatis.type-aliases-package = com.ndpmedia.base.entity
9 mybatis.mapper-locations = classpath:mapper/*.xml
10 mybatis.type-handlers-package = com.ndpmedia.mybatis
11 gears.server.host = ${GEARS_SERVER_HOST}
12 gears.server.domain = http://host.docker.internal:8091
13 app.datasource.first.driver-class-name = com.mysql.jdbc.Driver
14 app.datasource.first.jdbc-url = jdbc:mysql://${gears.db.host}:${gears.db.port}/${
15 app.datasource.first.username = ${gears.db.username}
16 app.datasource.first.password = ${gears.db.password}
17 app.datasource.v4.driver-class-name = com.mysql.jdbc.Driver
18 app.datasource.v4.jdbc-url = jdbc:mysql://${gears.db.host}:${gears.db.port}/${gears
19 app.datasource.v4.username = ${gears.db.username}
20 app.datasource.v4.password = ${gears.db.password}
21 logging.level.org.hibernate.SQL = info
22 logging.level.org.hibernate.type.descriptor.sql.BasicBinder = trace
23 spring.activemq.brokerUrl = tcp://host.docker.internal:61616
24 spring.activemq.broker-url = ${spring.activemq.brokerUrl}
25 spring.activemq.user = admin
26 spring.activemq.password = gears@123
27 mq.topic.sendmail = gears.topic.sendmail
28 mq.queue.sendmail = gears.queue.sendmail
29 spring.redis.host = host.docker.internal
30 spring.redis.password = gears@123
31 spring.redis.port = 6379
32 spring.redis.database = 0
33 spring.redis.timeout = 60s
34 spring.redis.jedis.pool.max-idle = 8
35 spring.redis.jedis.pool.min-idle = 0
36 spring.redis.jedis.pool.max-wait = -1s
37 spring.redis.jedis.pool.max-active = -1
38 gears.jenkins.password = 1111111
39 gears.jira-service.api = https://gears-service-dev.XXXXX.com/gears-jira/api
40 gears.jira-service.token = gears_jira_007
41 mq.queue.deploy = gears.queue.deploydev
42 mq.queue.log = gears.queue.logdev
43 X-TOKEN = la2p6F3uLsu0R77lao0RgWf3q
44 steampipe-api-gateway.base_url = http://steampipe-api-gateway.steampipe-dev.xxx.
45 steampipe-api-gateway.client_id = gears
46 steampipe-api-gateway.client_secret = 111111

```

## 8. Release the configuration.



## Step 8: Start Gears

1. Run the following command to start the Gears system:

```
1 ./start.sh
```

2. Wait for the system to initialize, which usually takes 1-2 minutes.

## Step 9: Access the Gears System

1. Open a web browser and enter the instance's public IP address followed by port 80, e.g., `http://<public_IP_of_instance>:80`.
2. If it's your first time accessing the system, use the initial administrator credentials:
  - Username: `admin@example.com`
  - Password: `admin@devops`