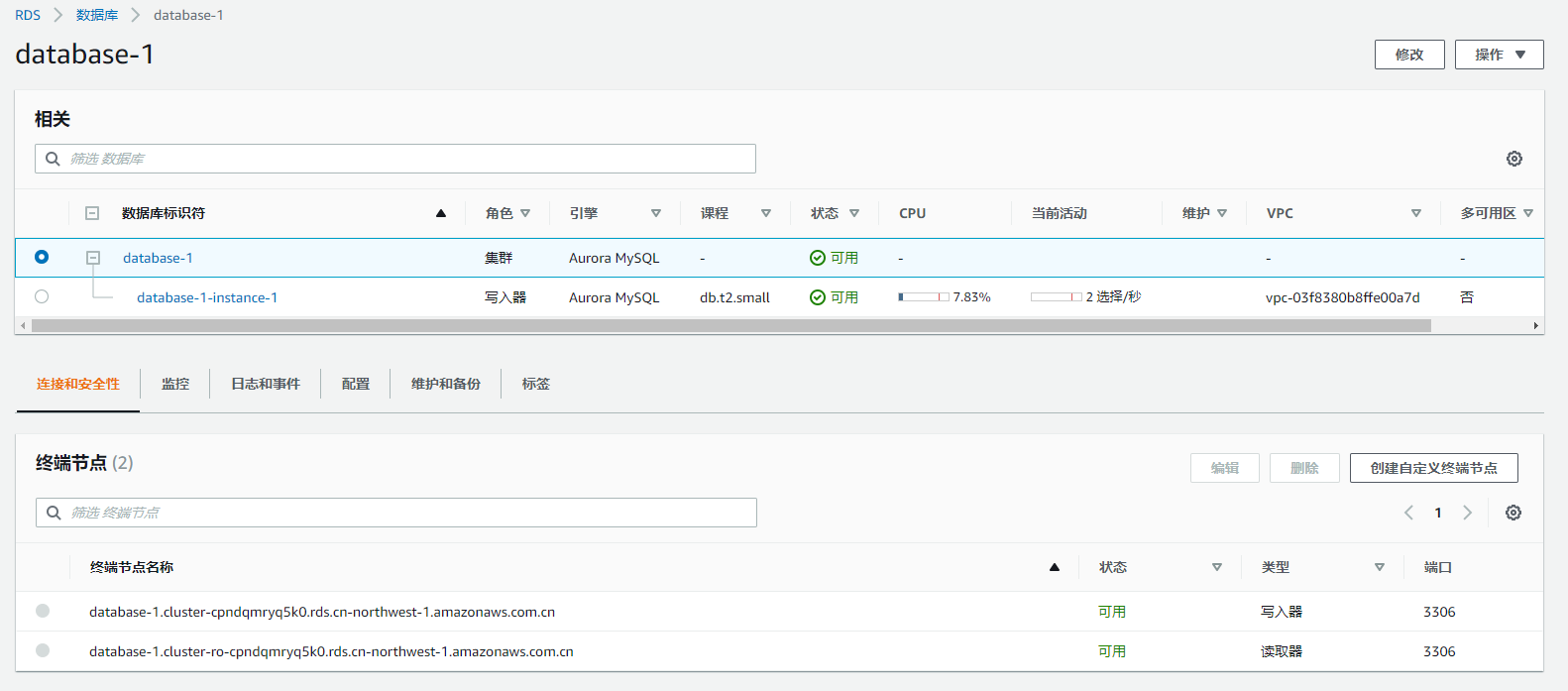
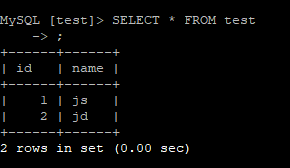
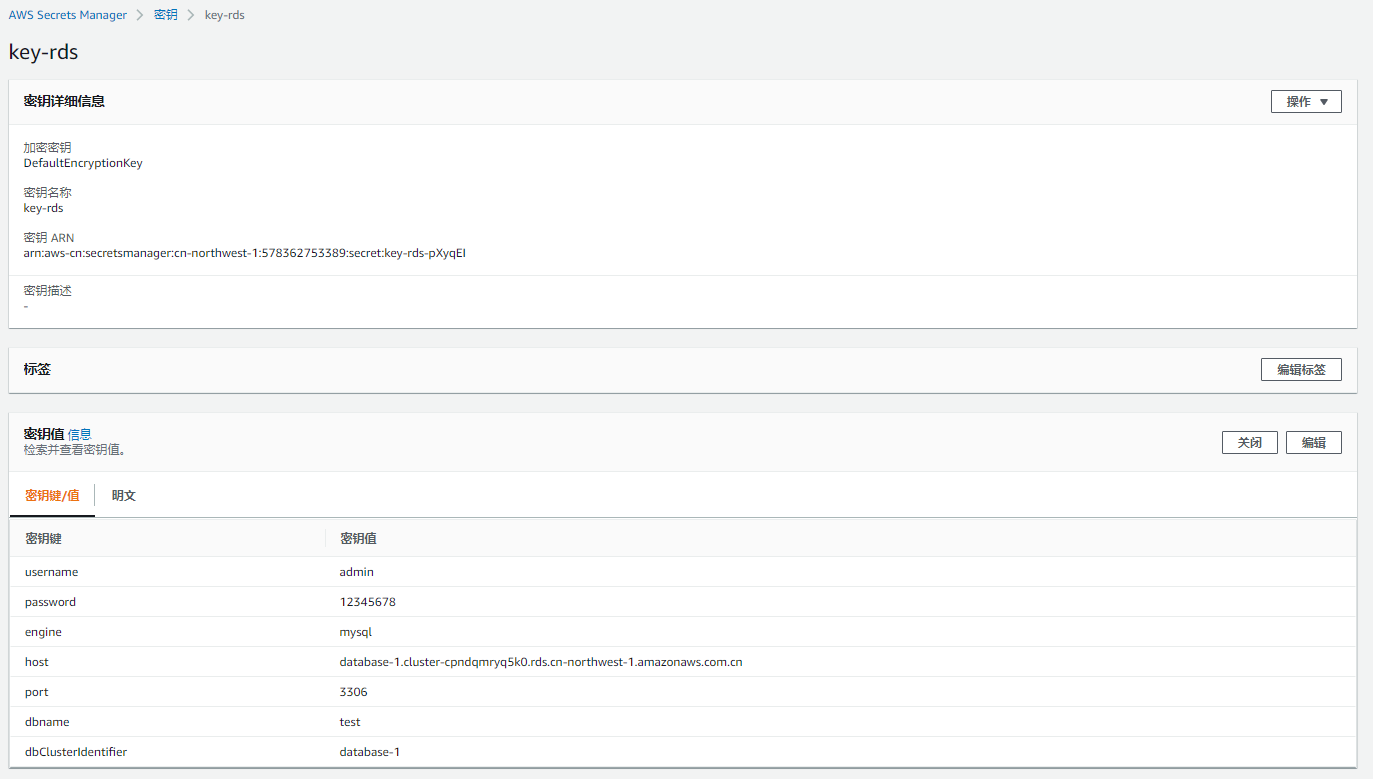
1.在RDS创建一个数据库



2.在数据库里创建表，插入测试数据



3.在secret manager里保存数据库用户名和密码



4.创建lambda层

因为lambda要登陆数据库获取数据，所以需要pymysql包

5.创建lambda函数，给IAM赋予相应的权限

1. **import** sys
2. **import** boto3
3. **import** botocore
4. **import** json
5. **import** pymysql
6. **import** os
8. **from** botocore.exceptions **import** ClientError
10. table\_name = os.environ['TB\_NAME']
11. db\_name = os.environ['RDS\_DB\_NAME']

14. #获取密钥
15. **def** get\_secret():
17. secret\_name = "key-rds"
18. region\_name = "cn-northwest-1"
20. # Create a Secrets Manager client
21. session = boto3.session.Session()
22. client = session.client(
23. service\_name='secretsmanager',
24. region\_name=region\_name
25. )
27. **try**:
28. get\_secret\_value\_response = client.get\_secret\_value(
29. SecretId=secret\_name
30. )
31. **except** ClientError as e:
32. **if** e.response['Error']['Code'] == 'DecryptionFailureException':
33. # Secrets Manager can't decrypt the protected secret text using the provided KMS key.
34. # Deal with the exception here, and/or rethrow at your discretion.
35. **raise** e
36. **elif** e.response['Error']['Code'] == 'InternalServiceErrorException':
37. # An error occurred on the server side.
38. # Deal with the exception here, and/or rethrow at your discretion.
39. **raise** e
40. **elif** e.response['Error']['Code'] == 'InvalidParameterException':
41. # You provided an invalid value for a parameter.
42. # Deal with the exception here, and/or rethrow at your discretion.
43. **raise** e
44. **elif** e.response['Error']['Code'] == 'InvalidRequestException':
45. # You provided a parameter value that is not valid for the current state of the resource.
46. # Deal with the exception here, and/or rethrow at your discretion.
47. **raise** e
48. **elif** e.response['Error']['Code'] == 'ResourceNotFoundException':
49. # We can't find the resource that you asked for.
50. # Deal with the exception here, and/or rethrow at your discretion.
51. **raise** e
52. **else**:
53. # Decrypts secret using the associated KMS CMK.
54. # Depending on whether the secret is a string or binary, one of these fields will be populated.
55. **if** 'SecretString' **in** get\_secret\_value\_response:
56. secret = get\_secret\_value\_response['SecretString']
57. **return** secret
58. **else**:
59. decoded\_binary\_secret = base64.b64decode(get\_secret\_value\_response['SecretBinary'])
60. **return** decoded\_binary\_secret


64. **def** lambda\_handler(event, context):
65. # TODO implement
66. content={}
67. secret=get\_secret()
68. SecretString=json.loads(secret)
69. user=SecretString['username']
70. pwd=SecretString['password']
71. host=SecretString['host']
72. #连接数据库
73. db = pymysql.connect(host,user,pwd,db\_name )
74. cursor = db.cursor()
75. #数据库语句
76. sql = "SELECT \* FROM "+table\_name
77. **try**:
78. # 执行SQL语句
79. cursor.execute(sql)
80. # 获取所有记录列表
81. results = cursor.fetchall()
82. **except**:
83. **print** ("Error: unable to fetch data")
84. **for** item **in** results:
85. kv=list(item)
86. content.update({kv[0]:kv[1]})
88. # 关闭数据库连接
89. db.close()
91. **return** {
92. 'statusCode': 200,
93. 'body': json.dumps('Hello from Lambda!'),
94. 'db':db\_name,
95. 'table\_name':table\_name,
96. 'content':json.dumps(content)

99. }

6.测试

