

WenQuanYi Micro Hei [Scale=0.9]WenQuanYi Micro Hei Mono song-  
WenQuanYi Micro Hei sfWenQuanYi Micro Hei "zh" = 0pt plus 1pt

---

# **KaliArch's BLOG**

***Release 1.0.0***

**Oct 31, 2018**

---

# Python a Odpor ECS e G l a l i e T I J a C R a L Z a z z e T I J a C R a d n a L u e G s a E u a z

---

<b>1</b>	<b>äyÄãÄÄèÇÑæŽr</b>	<b>2</b>
1.1	1.1 éŮóécŸijŽ . . . . .	2
1.2	1.2 æÄIèúřijŽ . . . . .	2
<b>2</b>	<b>äžŇãÄÄäžččäÄ</b>	<b>3</b>
2.1	2.1 çzŠæđĎ . . . . .	3
2.2	2.2 æäyâĚČäžččäÄ . . . . .	3
<b>3</b>	<b>äyL'ãÄÄætŇerŤ</b>	<b>7</b>
3.1	3.1 æšëçIJŇèĚŘèqŇçzŠæđIJ . . . . .	7
3.2	3.2 æšëçIJŇwebæŮgãLúãŘř . . . . .	7
3.3	3.3 æšëçIJŇæŮèâĚŮ . . . . .	9
<b>4</b>	<b>ãžŽãÄÄäijYãŇŮ</b>	<b>10</b>
<b>5</b>	<b>äyÄãÄÄèÇÑæŽr</b>	<b>11</b>
<b>6</b>	<b>äžŇãÄÄæĚÄæIJrçzEèĚĚ</b>	<b>12</b>
<b>7</b>	<b>äyL'ãÄÄäžččäÄ</b>	<b>13</b>
7.1	3.1 éĚ■ç;óæŮĚgãžú . . . . .	13
7.2	3.2 CDB APIæäyâĚČæŠ■ä;IJäžččäÄ . . . . .	14
7.3	3.3 COS SDKæäyâĚČæŠ■ä;IJäžččäÄ . . . . .	15
7.4	3.4 æŮèâĚŮèõřã;ŤæäyâĚČäžččäÄ . . . . .	15
<b>8</b>	<b>ãžŽãÄÄætŇerŤçzŠæđIJ</b>	<b>17</b>
<b>9</b>	<b>äžŤãÄÄæÄžçzŠ</b>	<b>18</b>

## Contents

- *äyÄãÄÄèČŇæŽř*
  - 1.1 *éŮóécŸüijŽ*
  - 1.2 *æÄIèürüijŽ*
- *äžŇãÄÄäzčçäÄ*
  - 2.1 *çzšæđĎ*
  - 2.2 *æÿÿåŁČäzčçäÄ*
- *äyLãÄÄæŧŇèřŧ*
  - 3.1 *æšěçIJŇèŁŘèqŇçzšæđIJ*
  - 3.2 *æšěçIJŇwebæŎğåŁúåŘř*
  - 3.3 *æšěçIJŇæŮěåŁŮ*
- *äžžãÄÄäijŸäŇŮ*

# CHAPTER 1

---

äÿĂăĀÀèĈŇæŽř

---

## 1.1 1.1 éUóécŸiijŽ

âŔŇăžŇâŔ■éçŁæIJL'âŔřăžééL't'ăžŎçŽóâL'■âĜăæñâd'ğçŽĎăĚñæIJL'ăžSăžŇæŦĚiijŇèĚ;èóřăžŚ/éŸŁé

## 1.2 1.2 æĀlèùríijŽ

âIJlæIJĀâd'ğçlŇăžççŽĎéŽ■ă;ŎITæĹŔæIJñiijŇâŔĹæĈşâIJläÿ■âŔŕæŎğâd'ğèğĎæłâIJřâşşæĀğçA;éŽ;é

# CHAPTER 2

## 2.1 2.1 配置目录

### 2.1 2.1 配置目录



github.com/51cto

### 2.2 2.2 配置目录

> 配置目录

```
# [common]
# 配置目录
accessKeyId = LTAIhfX1cjyln6tW
# 配置目录
accessSecret = GwfAMvR4K2ELmt76184oqLTVgRfAso
# 配置目录
logdir_name = logdir
```

(continues on next page)

(continued from previous page)

```

# logæŰĞăzúăŘ■çğř
logfile_name = ecsoperlog.log

# ecsæžŘăIJřăSSéĚ■ç;ôăĤăæĀřăōț
#ăŤřăŇăăIJlă■ÓăŇŮ 1ăĂăă■ÓăŇŮ 2ăĂăă■ÓăŇŮ 3ăĂăă■ÓăŇŮ 5ăĂăă■ÓăŷIJ
→1ăĂăă■ÓăŷIJ 2 äŖŇă■ÓăŮ 1
→ăIJřăSSăžŇéŮt'äd'■ăĹúéŤIJăČŘăĂĆăúĹ'ăŘĹăĚúăžŮăž;ăóúăŖŇăIJřăŇžăIJřăSSăŮúii jŇăŘă
→ăŘăžď'ăúěă■Ť çŤșèrú
[source]
# æžŘăIJřăSSăôďă;Ňregionidii jŇăŘăžăăŘĆăĂĆii jžhttps://help.aliyun.
→com/document_detail/40654.html?spm=a2c1g.8271268.10000.5.
→5f98df25B98bhJ
s_RegionId = cn-shanghai

# æžŘăôďă;Ňid,ăŘăŇăăŮăōžăđ'žăŷlçŤĹii jŇèĤžăăŇăĹĚéžŤ
s_InstanceId = i-uf661wb708uvqc9jyhem,i-uf661wb708uvqc9jyhel

# æžŘçŇăăĹúă;IJéŤIJăČŘname
s_ImageName = api-source-image

# æžŘéŤIJăČŘăŘĚèĤřăĤăæĀř
s_Description = api-source-imageæžŘéŤIJăČŘăŘĚèĤřăĤăæĀř

# éŤIJăČŘăđ'■ăĹúçžŮçžĐăIJřăSSéĚ■ç;ôăĤăæĀřăōț
[destination]
# çžŮçžĐăIJřăSSăôďă;Ňregionidii jŇ
d_DestinationRegionId = cn-qingdao

# äđ'■ăĹúèĤĞăĲéçžĐéŤIJăČŘăăŘ■çğř
d_DestinationImageName = api-destination-image

# äđ'■ăĹúèĤĞăĲéçžĐéŤIJăČŘăŘĚèĤřăĤăæĀř
d_DestinationDescription = api-destination-
→imageçžŮçžĐéŤIJăČŘăŘĚèĤřăĤăæĀř

imageæŖ■ă;IJ(ăĹúă;IJéŤIJăČŘ→æŖçIJŇéŤIJăČŘăĹúă;IJçĹúăĂă→
→äd'■ăĹúéŤIJăČŘ)

```

```

# äĹžăžžăôďă;ŇçŤŖăĹăăžĹ
def _get_Instance(self):
    for Instance in self.s_InstanceId_list.split(','):
        yield Instance

def _create_image(self):
    """
    äĹžăžžéŤIJăČŘ
    :return:èĤŤăžďéŤIJăČŘid
    """
    s_timer = time.strftime("%Y-%m-%d-%H:%M", time.
→localtime(time.time()))

```

(continues on next page)

(continued from previous page)

```

request = CreateImageRequest.CreateImageRequest ()
request.set_accept_format ('json')
request.add_query_param ('RegionId', self.s_RegionId)
request.add_query_param ('InstanceId', self.s_InstanceId)
request.add_query_param ('ImageName', self.s_ImageName + s_
↪timer)
request.add_query_param ('Description', self.s_Description +
↪s_timer)
response = self.ecshelper.do_action_with_exception(request)
self.logoper.info ('áĹžážžéŤIJáČŘážžáĹāāũšæŘŘážd', éŤIJáČŘid:%s
↪' % json.loads(response) ["ImageId"])
print ('áĹžážžéŤIJáČŘážžáĹāāũšæŘŘážd', éŤIJáČŘid:%s' % json.
↪loads(response) ["ImageId"])
return json.loads(response) ["ImageId"]

def _describe_image(self, imageid):
    """
    æŠëèřćimageçĹúæĀĀ
    :param imageid:
    :return:
    """
    request = DescribeImagesRequest.DescribeImagesRequest ()
    request.set_accept_format ('json')
    request.add_query_param ('RegionId', self.s_RegionId)
    request.add_query_param ('ImageId', imageid)
    response = self.ecshelper.do_action_with_exception(request)
    # èĹžážžè json.loads(response) ['Images'] ['Image'] [0] ['Progress
↪']
    self.logoper.info ('éŤIJáČŘáĹžážžèĹžážžè:%s' % json.
↪loads(response) ['Images'] ['Image'] [0] ['Progress'])
    # éŤIJáČŘçĹúæĀĀ
    return json.loads(response) ['Images'] ['Image'] [0] ['Status']

#éŤIJáČŘád'■áĹŮ
def _copy_image(self, imageid):
    """
    éŤIJáČŘád'■áĹŮ
    :param imageid:æžŘéŤIJáČŘid
    :return: ád'■áĹŮæĹŮáĹŮáĹŮáĹŮáĹŮáĹŮáĹŮáĹŮáĹŮáĹŮ
    """
    flag = True
    while flag:
        try:
            if self._describe_image(imageid) == 'Available':
                flag = False
            else:
                time.sleep(300)
        except Exception as e:

```

(continues on next page)



(continued from previous page)

```

    pass
    print('éŤIJăČŘăŭščžŘăĹžăžžăóŇăĹŘ')
    d_timer = time.strftime("%Y-%m-%d-%H:%M", time.
↳ localtime(time.time()))
    request = CopyImageRequest.CopyImageRequest()
    request.set_accept_format('json')
    request.add_query_param('RegionId', self.s_RegionId)
    request.add_query_param('DestinationRegionId', self.d_
↳ DestinationRegionId)
    request.add_query_param('DestinationImageName', self.d_
↳ DestinationImageName + d_timer)
    request.add_query_param('DestinationDescription', self.d_
↳ DestinationDescription + d_timer)
    request.add_query_param('ImageId', imageid)
    response = self.ecshelper.do_action_with_exception(request)
    self.logoper.info('ăđ'■ăĹŭéŤIJăČŘăžžăĹăăŭšăŘăžđ', éŤIJăČŘid: %s
↳ ' % json.loads(response) ['ImageId'])
    print('ăđ'■ăĹŭéŤIJăČŘăžžăĹăăŭšăŘăžđ', éŤIJăČŘid: %s' % json.
↳ loads(response) ['ImageId'])
    return json.loads(response) ['ImageId']

```

### 3.1 创建镜像任务

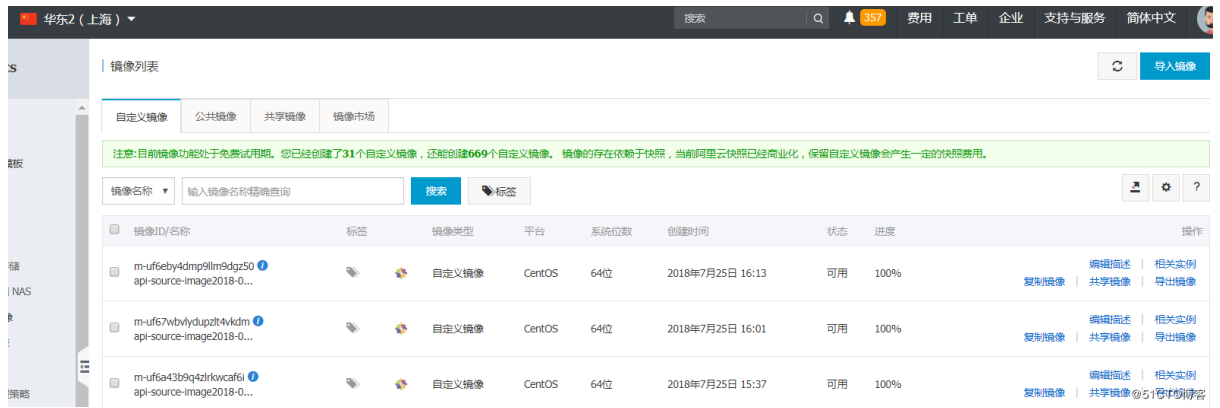
```
create_cop_image_new x
D:\ProgramData\Anaconda3\envs\python36\python.exe D:\pythonwork\Pyworkstatio
实例i-uf661wb708uvqc9jyhem,创建镜像任务已提交,镜像id:m-uf6cp52t42s72i4jgqli
镜像已经创建完成
复制镜像任务已提交,镜像id:m-m5e8b39oaqnhbfp9x35i
实例i-uf661wb708uvqc9jyhel,创建镜像任务已提交,镜像id:m-uf60h59flke4cqhtg6nn
镜像已经创建完成
复制镜像任务已提交,镜像id:m-m5ednmzrres9mkjjsq0l

Process finished with exit code 0
```

@51CTO博客

### 3.2 镜像任务web

> 创建镜像任务



> æúãŁääžEæÛúéÛt'æŁšijÑæÛzä;£æšëçIJÑ

### 编辑自定义镜像描述

自定义镜像ID : m-uf6a43b9q4zlrqya2b88

自定义镜像名称 : api-source-image2018-0...

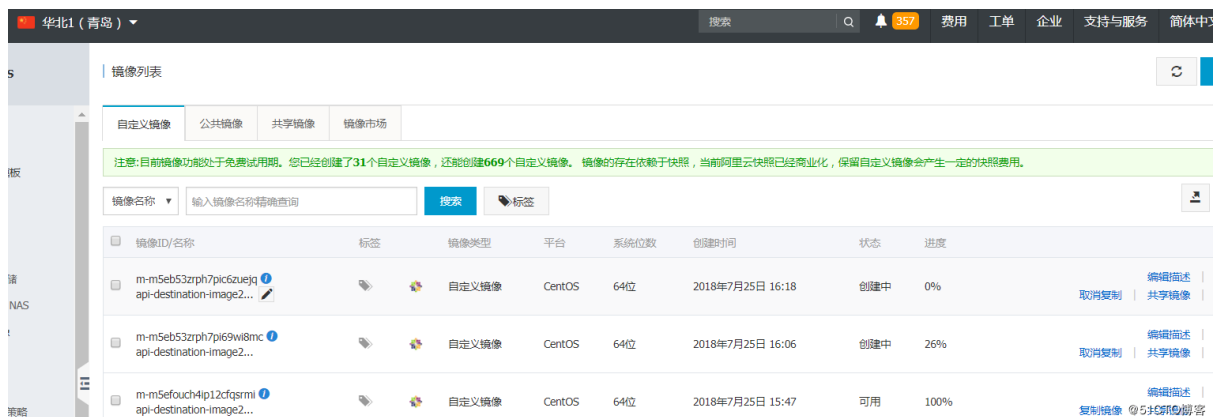
\* 自定义镜像描述 :

长度为2-256个字符，不能以http://或https://开头。

标签 :

@51CTO博客

> çZöçŽDâIJrãššëçIJÑãČR



### 3.3 3.3 æšëçlJÑæÛěå£Û

```
2018-07-26 10:16:59,150 - root - INFO - 创建镜像任务已提交,镜像id:m-uf6cp52t42s72i4jgqli
2018-07-26 10:22:00,824 - root - INFO - 镜像创建进度:100%
2018-07-26 10:22:01,314 - root - INFO - 复制镜像任务已提交,镜像id:m-m5e8b39oaqnhbfp9x35i
2018-07-26 10:22:03,383 - root - INFO - 创建镜像任务已提交,镜像id:m-uf60h59flke4cqhtg6nn
2018-07-26 10:27:04,725 - root - INFO - 镜像创建进度:100%
2018-07-26 10:27:05,060 - root - INFO - 复制镜像任务已提交,镜像id:m-m5ednmzrres9mkjjsq0l
```

@51CTO博客

## CHAPTER 4

---

### åŽŽãÄÄäijÿãŇÚ

---

- åŔřãžãŔŔŔz■áçđãĽããřãæŇĜãóŽãd'ŕ æŦřçŽĐéŦIJãČŔè£ŽèãŇã;ŠæãçãĽãéŽd'

#### Contents

- äÿÄãÄÄèČŇæŽř
- äžŇãÄÄæĽÄæIJřçžEèĽĆ
- äÿĽãÄÄäžççãÄ
  - 3.1 éĚ■ç;ŕæŮĜäžŮ
  - 3.2 CDB APIæäÿã£ČæŠ■ä;IJäžççãÄ
  - 3.3 COS SDKæäÿã£ČæŠ■ä;IJäžççãÄ
  - 3.4 æŮèã£Ůèŕã;Ŧæäÿã£ČäžççãÄ
- åŽŽãÄÄæŦŇèřŦçžçæđIJ
- äžŦãÄÄæÄžççç

## CHAPTER 5

---

### äÿÄãÄÀèĈŇæŽř

---

- éIJĀæsĆiijŽčŽóáL'■éAĞáLřčŽĎáóćæLúéIJĀæsĆäyžârEèĚèóřázŚCDBåd' Ğäz;æŰĜäzúèĜlâLlâyLäijä
- æÄIèürriijŽéçŰáĚLèŎuâRŰáLřCDBçŽĎád' Ğäz;äyŇè;ıurlriijŇéĂŽè£ĜCOSçŽĎAPIäyLäijäæŰĜäzúriij

# CHAPTER 6

## äžÑãÄAæŁÄæIJrçzEèŁĆ

- COSiijŽCOSæIJL'APIãÑæUúæIJL'SDKiijÑæŁZãrsã;ŁæÚzã;ŁæŁSãžñæÌééÄŽèŁĞPythonãrZCOSèŁZ SDK for Python
- CDBiijŽCDBæIJL'APIã;EæÝrCDBçŽĐæšèèrcãd'Gãz;ãýÑè;æšææIJL'ãrZãžTçŽĐSDKiijÑæd'æUúãÑæ >ãrZdictæŌŠãžR->æÑijæŌèsign->ãrZsignçijÚçãA->æÑijæŌèãŌÑæLRæIJãçzLurl->ãŌÑæLRèrČçTliijÑç;ãRæÚzæšTiiijÑæšèèrcãd'Gãz;API
- requirements:

```
cos-python-sdk-v5==1.5.2
requests==2.19.1
tencentcloud-sdk-python==3.0.15
urllib3==1.23
```

▼ cdbbak_to_cos	日志目录
▼ rds_to_cos	日志文件
2018-07-16-rdsbak_to_cos.log	cos/cdb/log 配置文件
config.txt	cdb操作文件
getcdbbakurl.py	log操作文件
loghelper.py	主函数调用
main.py	requirements文件
requirements.txt	cos操作文件
uploadcos.py	正在完成上传的cdb备份文件
冷备_backup_20180716060158	

- æÚGãžúçŽŌã;TçzŠædĐ

# CHAPTER 7

## äyL'ãÄÄzčçäA

githubâIJrãÄ

### 7.1 3.1 éĚ■ç;õæÜĜäzú

```
# auth:kaliarch
# func:ãřĚèĚ;èõrãžšcldbãd' Ĝäz;æŮĜäzŮäyŁäijäèĜşcosãLúãóžçžĎbucketãĚĚ
# python version:python3+
# cos version:v5
# https://console.cloud.tencent.com/cos5/bucket

# èĚ;èõrãžšãĚñãĚsãfæAřéĚ■ç;õæõť
[common]
# èĚ;èõrãžš secretid
secret_id = AKIDMdjegcmoGxxxxxxxxxxxxxxxxxxxxxxx
# èĚ;èõrãžš secretkey
secret_key = d5MRL4Voxyv1Qxxxxxxxxxxxxxxxxxxxxxxx

# èĚ;èõrãžšcosãfæAřéĚ■ç;õæõť
[cosinfo]
# cosãL'ÄãIJÍãIJrãšš
cos_region = ap-chengdu

# èĚ;èõrãžšbucketãĚ■ãŮ(cos v5 bucketãĚ■çğřçžĎãĹĚ:bucket+appid)
bucket_name = xuel-test-bucket-125396xxxx

# èĚ;èõrãžšcldbãfæAřéĚ■ç;õæõť
[cdbinfo]
```

(continues on next page)



(continued from previous page)

```

# cdbāōđä;Ñid
cdb_instanceid = cdb-rqaxxxxx

# cdbæL'ĀāIJlāIJřāS$
cdb_region = ap-shanghai

# cdb_
↳æŮěāŁŮād' Ğäz;çszādÑiijÑcoldbackupiijLāEūād' ĞiijL' iijÑbinlogiijLāžÑèŁZāLūæŮěāŁŮ
↳dayiijlāĒcæSèèércæŮěāŁŮiijL'
cdb_bak_type = coldbackup

# æŮěāŁŮæŮĜäzūäŁæAréĒç;ōæōŁ
[loginfo]
#æŮěāŁŮæŮĜäzūçZōā;ŤāŘçğř
logdir_name = rds_to_cos
#æŮěāŁŮæŮĜäzūāŘçğř
logfile_name = rdsbak_to_cos.log

```

## 7.2 3.2 CDB APIæăÿāŁČæŞ■ä;IjäžččăĀ

```

#æđĎāžžā■ŮāĚÿ
keydict = {
    'Action': self.cdb_action,
    'Timestamp': str(int(time.time())),
    'Nonce': str(int(random.random() * 1000)),
    'Region': self.cdb_region,
    'SecretId': self.secret_id,
    # 'SignatureMethod': SignatureMethod,
    'cdbInstanceId': self.cdb_instanceid,
    'type': self.cdb_bak_type
}
#ā■ŮāĚÿæŮŌšāžŘ
sorted(zip(keydict.keys(), keydict.values()))
#ā■ŮçņęăÿšæÑijæŮě
sign_str_init = ''
for value in sortlist:
    sign_str_init += value[0] + '=' + value[1] + '&'
sign_str = 'GET' + self.cdb_api_url + sign_str_init[:-1]
return sign_str, sign_str_init
#èŮūāŘŮç■;āŘ■ăÿšāžŮçijŮçăĀ
secretkey = self.secret_key
signature = bytes(sign_str, encoding='utf-8')
secretkey = bytes(secretkey, encoding='utf-8')
my_sign = hmac.new(secretkey, signature, hashlib.sha1).digest()
my_sign = base64.b64encode(my_sign)
parse.quote(my_sign)

```

(continues on next page)

(continued from previous page)

```
#èÕùáRÛæIJĂçžŁurl
result_url = 'https://' + self.cdb_api_url + sign_str + '&Signature='
↳ ' + result_sign
```

â■ŤçŃñèŁRèàŃæ■d' æłāāIŪāRřázěā;ŪāLřázěäyŃāŁæAřijŽ

```
getcdbbakurl
D:\ProgramData\Anaconda3\envs\python36\python.exe D:/pythonwork/Pyworkstation/cdb-cos/cdbbak_to_cos/getcdbbakurl.py
cdbbak_name: 冷备_backup_20180716060158,
cdb_download_url: https://sh-dl-cdb.qcloud.com/5ff2a7d11dd677d13f50bb036c52645e?appid=1253961596&time=1531707978&sign=KFWgPCIdv
cdb_download_size:,656M

Process finished with exit code 0
```

### 7.3 3.3 COS SDKæäyåŁČæŞ■ä;IjäžčçăA

```
↳ #æžæ■ōæŪĞäzúâd' ğāřRèĞłāŁléĀL' æŃl' çŌĀâ■ŤäyŁäijæLŪāLĒāIŪäyŁäijäiijŃāŁæAřijŽ
with open(filename, 'wb') as localfile:
    localfile.write(requests.request('get', url).content)
# èŁžèàŃäyŁäijă
response = cos_client.upload_file(
    Bucket=self.bucket_name,
    LocalFilePath=filename,
    Key=filename,
    PartSize=partsize,
    MAXThread=maxthread
)
# āLăéžd' æIŃñâIŤřæŪĞäzú
if os.path.exists(filename):
    os.remove(filename)
```

### 7.4 3.4 æŪěåŁŪèőřā;ŤæäyåŁČäzčçăA

```
#āLžāžžçžŌā;Ť
def create_dir(self):
    _LOGDIR = os.path.join(os.path.dirname(__file__), self.
↳ logdir_name)
    _TIME = time.strftime('%Y-%m-%d', time.gmtime()) + '-'
    _LOGNAME = _TIME + self.logfile_name
    LOGFILENAME = os.path.join(_LOGDIR, _LOGNAME)
    if not os.path.exists(_LOGDIR):
        os.mkdir(_LOGDIR)
    return LOGFILENAME
#āŌžžāzL' æŪěåŁŪæŪĞäzú
def create_logger(self, logfile_name):
    logger = logging.getLogger()
    logger.setLevel(logging.INFO)
```

(continues on next page)

(continued from previous page)

```
handler = logging.FileHandler(logfilename)
handler.setLevel(logging.INFO)
formater = logging.Formatter('%(asctime)s - %(name)s -
↪%(levelname)s - %(message)s')
handler.setFormatter(formater)
logger.addHandler(handler)
return logger
```

# CHAPTER 8

## åZZãÄAætÑèrTçzŞædlJ

èÕúâRÚCDBäyÑè;ijéŞ;æÕě

áoÑæLŘäyLäijäæşëçIJNCOSæŪĜazú

备份文件	开始时间	结束时间	备份大小	类型	备份方式
冷备_backup_20180716060158	2018-07-16 06:01:58	2018-07-16 06:02:28	656 MB	逻辑冷备	自动
冷备_backup_20180715030620	2018-07-15 03:06:20	2018-07-15 03:06:32	656 MB	逻辑冷备	自动
冷备_backup_20180715030038	2018-07-15 03:00:38	2018-07-15 03:00:56	656 MB	逻辑冷备	自动
冷备_backup_20180714020703	2018-07-14 02:07:03	2018-07-14 02:07:13	656 MB	逻辑冷备	自动

文件名	大小	存储类型
Huangfh_backup_20180716045117	1.05MB	标准存储
冷备_backup_20180716060158	656.33MB	标准存储

## CHAPTER 9

---

### äžTãÄAæÄžçžŞ

---

- äijŸãÑŮ:ãRřazěãRŮæIJšéÄŽè£GéĚ■ãRĹãóŽæŮŮäzzãŁãóŃæLŘèGłãŁãÑŮäzzãŁã
- æLŮãšŤ:æžRçñriijžäy■äžĚäžĚãšÄéŽRäžŮCDBãd' Ğäz;æŮĞäžŮiiijŃärzäžŮéŽRä;£äyŃè;ıurliijŃãIĞã  
S3,çŽ;ãžçäžŞBOS ç■L'ãĀĆ