

# Aliyun Elastic IP の管理

このスクリプトは、Aliyun Elastic IP (EIP) を管理するためのコマンドラインインターフェースを提供します。Aliyun SDK for Python を使用して、EIP の作成、バインド、アンバインド、解放を行うことができます。スクリプトは、実行するジョブと EIP の割り当て ID を引数として受け取ります。

```
python aliyun_elastic_ip_manager.py unbind --allocation_id eip-j6c2olvs7jk9142iaaa
python aliyun_elastic_ip_manager.py bind --allocation_id eip-j6c7mhenamvy6zao3haaa
python aliyun_elastic_ip_manager.py release --allocation_id eip-j6c2olvs7jk9142aaa
python aliyun_elastic_ip_manager.py describe

# -*- coding: utf-8 -*-

# このファイルは自動生成されたものです。編集しないでください。

import logging
import os
import sys
from typing import List
import argparse
import json

from alibabacloud_vpc20160428.client import Client as Vpc20160428Client
from alibabacloud_tea_openapi import models as open_api_models
from alibabacloud_vpc20160428 import models as vpc_20160428_models
from alibabacloud_tea_util import models as util_models
from alibabacloud_tea_util.client import Client as UtilClient
from alibabacloud_ecs20140526.client import Client as Ecs20140526Client

logging.basicConfig(level=logging.INFO, format='%(asctime)s - %(levelname)s - %(message)s')

class Sample:
    def __init__(self):
        pass

    @staticmethod
    def create_client() -> Vpc20160428Client:
        config = open_api_models.Config(
            access_key_id=os.environ['ALIBABA_CLOUD_ACCESS_ID_API_KEY'],
            access_key_secret=os.environ['ALIBABA_CLOUD_ACCESS_API_KEY']
        )
```

```

config.endpoint = f'vpc.cn-hongkong.aliyuncs.com'
return Vpc20160428Client(config)

@staticmethod
def bind_eip(
    region_id: str,
    allocation_id: str,
    instance_id: str,
) -> bool:
    client = Sample.create_client()
    associate_eip_address_request = vpc_20160428_models.AssociateEipAddressRequest(
        region_id=region_id,
        allocation_id=allocation_id,
        instance_id=instance_id
    )
    runtime = util_models.RuntimeOptions(read_timeout=60000, connect_timeout=60000)
    try:
        result = client.associate_eip_address_with_options(associate_eip_address_request, runtime)
        logging.info(f"EIP {allocation_id} をインスタンス {instance_id} に正常にバインドしました。結果: {result}")
        return True
    except Exception as error:
        logging.error(f"EIP {allocation_id} をインスタンス {instance_id} にバインドする際にエラーが発生しました。エラーメッセージ: {error.message}")
        if hasattr(error, 'data') and error.data and error.data.get('Recommend'):
            logging.error(f" 推奨事項: {error.data.get('Recommend')}")
        UtilClient.assert_as_string(str(error))
        return False

@staticmethod
def unbind_eip(
    region_id: str,
    allocation_id: str,
    instance_id: str,
) -> bool:
    client = Sample.create_client()
    unassociate_eip_address_request = vpc_20160428_models.UnassociateEipAddressRequest(
        region_id=region_id,
        allocation_id=allocation_id,
        instance_id=instance_id
    )

```

```

)
runtime = util_models.RuntimeOptions(read_timeout=60000, connect_timeout=60000)
try:
    result = client.unassociate_eip_address_with_options(unassociate_eip_address_request, runtime)
    logging.info(f"EIP {allocation_id} をインスタンス {instance_id} から正常にアンバインドしました。")
    return True
except Exception as error:
    logging.error(f"EIP {allocation_id} をインスタンス {instance_id} からアンバインドする際にエラーが発生しました: {error}")
    if hasattr(error, 'message'):
        logging.error(f" エラーメッセージ: {error.message}")
    if hasattr(error, 'data') and error.data and error.data.get('Recommend'):
        logging.error(f" 推奨事項: {error.data.get('Recommend')}")
    UtilClient.assert_as_string(str(error))
    return False


@staticmethod
def create_eip(
    region_id: str,
) -> str | None:
    client = Sample.create_client()
    allocate_eip_address_request = vpc_20160428_models.AllocateIpAddressRequest(
        region_id=region_id
    )
    runtime = util_models.RuntimeOptions(read_timeout=60000, connect_timeout=60000)
    try:
        result = client.allocate_eip_address_with_options(allocate_eip_address_request, runtime)
        print(result.body)
        allocation_id = result.body.allocation_id
        logging.info(f"EIP を正常に作成しました。割り当て ID: {allocation_id}")
        return allocation_id
    except Exception as error:
        logging.error(f"EIP の作成中にエラーが発生しました: {error}")
        if hasattr(error, 'message'):
            logging.error(f" エラーメッセージ: {error.message}")
        if hasattr(error, 'data') and error.data and error.data.get('Recommend'):
            logging.error(f" 推奨事項: {error.data.get('Recommend')}")
        UtilClient.assert_as_string(str(error))
        return None

```

@staticmethod

```

def release_eip(
    allocation_id: str,
) -> bool:
    client = Sample.create_client()
    release_eip_address_request = vpc_20160428_models.ReleaseEipAddressRequest(
        allocation_id=allocation_id
    )
    runtime = util_models.RuntimeOptions(read_timeout=60000, connect_timeout=60000)
    try:
        result = client.release_eip_address_with_options(release_eip_address_request, runtime)
        logging.info(f"EIP {allocation_id} を正常に解放しました。結果: {result}")
        return True
    except Exception as error:
        logging.error(f"EIP {allocation_id} の解放中にエラーが発生しました: {error}")
        if hasattr(error, 'message'):
            logging.error(f" エラーメッセージ: {error.message}")
        if hasattr(error, 'data') and error.data and error.data.get('Recommend'):
            logging.error(f" 推奨事項: {error.data.get('Recommend')}")
        UtilClient.assert_as_string(str(error))
        return False

@staticmethod
def describe_eip(
    region_id: str,
) -> None:
    client = Sample.create_client()
    describe_eip_addresses_request = vpc_20160428_models.DescribeEipAddressesRequest(
        region_id=region_id
    )
    runtime = util_models.RuntimeOptions(read_timeout=60000, connect_timeout=60000)
    try:
        result = client.describe_eip_addresses_with_options(describe_eip_addresses_request, runtime)
        logging.info(f"EIP の情報を正常に取得しました。")
        print(json.dumps(result.body.to_map(), indent=4))
    except Exception as error:
        logging.error(f"EIP の情報の取得中にエラーが発生しました: {error}")
        if hasattr(error, 'message'):
            logging.error(f" エラーメッセージ: {error.message}")
        if hasattr(error, 'data') and error.data and error.data.get('Recommend'):
            logging.error(f" 推奨事項: {error.data.get('Recommend')}")

```

```

UtilClient.assert_as_string(str(error))

@staticmethod
def main(
    args: List[str],
) -> None:
    region_id = "cn-hongkong"
    instance_id = "i-j6c44l4zpphv7u7agdbk"

    parser = argparse.ArgumentParser(description='Aliyun Elastic IP を管理します。')
    parser.add_argument('job', choices=['create', 'bind', 'unbind', 'release', 'describe'], help='実行する操作')
    parser.add_argument('--allocation_id', type=str, help='EIP の割り当て ID。')
    parser.add_argument('--instance_id', type=str, default=instance_id, help='EIP をバインド/アンバインドするインスタンス ID。')

    parsed_args = parser.parse_args(args)

    if parsed_args.job == 'create':
        new_allocation_id = Sample.create_eip(region_id)
        if new_allocation_id:
            print(f"EIP の作成プロセスが正常に開始されました。割り当て ID: {new_allocation_id}")
        else:
            print("EIP の作成プロセスが失敗しました。")
    elif parsed_args.job == 'bind':
        if not parsed_args.allocation_id:
            print(" エラー: bind ジョブには --allocation_id が必要です。")
            return
        if Sample.bind_eip(region_id, parsed_args.allocation_id, parsed_args.instance_id):
            print(f"EIP {parsed_args.allocation_id} とインスタンス {parsed_args.instance_id} のバインドが成功しました。")
        else:
            print(f"EIP {parsed_args.allocation_id} とインスタンス {parsed_args.instance_id} のバインドが失敗しました。")
    elif parsed_args.job == 'unbind':
        if not parsed_args.allocation_id:
            print(" エラー: unbind ジョブには --allocation_id が必要です。")
            return
        if Sample.unbind_eip(region_id, parsed_args.allocation_id, parsed_args.instance_id):
            print(f"EIP {parsed_args.allocation_id} とインスタンス {parsed_args.instance_id} のアンバインドが成功しました。")
        else:
            print(f"EIP {parsed_args.allocation_id} とインスタンス {parsed_args.instance_id} のアンバインドが失敗しました。")
    elif parsed_args.job == 'release':
        if not parsed_args.allocation_id:
            print(" エラー: release ジョブには --allocation_id が必要です。")
            return
        if Sample.release_eip(region_id, parsed_args.allocation_id):
            print(f"EIP {parsed_args.allocation_id} のリリースが成功しました。")
        else:
            print(f"EIP {parsed_args.allocation_id} のリリースが失敗しました。")
    elif parsed_args.job == 'describe':
        if not parsed_args.allocation_id:
            print(" エラー: describe ジョブには --allocation_id が必要です。")
            return
        allocation_id = parsed_args.allocation_id
        if allocation_id:
            allocation_info = Sample.describe_eip(allocation_id)
            print(f"EIP {allocation_id} の情報: {allocation_info}")
        else:
            print("EIP の情報が取得できませんでした。")
    else:
        print("不明な操作です。")

```

```

if not parsed_args.allocation_id:
    print(" エラー: release ジョブには --allocation_id が必要です。")
    return

if Sample.release_eip(parsed_args.allocation_id):
    print(f"EIP {parsed_args.allocation_id} の解放プロセスが正常に開始されました。")
else:
    print(f"EIP {parsed_args.allocation_id} の解放プロセスが失敗しました。")

elif parsed_args.job == 'describe':
    Sample.describe_eip(region_id)

@staticmethod
async def main_async(
    args: List[str],
) -> None:
    client = Sample.create_client()
    associate_eip_address_request = vpc_20160428_models.AssociateEipAddressRequest(
        region_id='cn-hongkong'
    )
    runtime = util_models.RuntimeOptions()
    try:
        await client.associate_eip_address_with_options_async(associate_eip_address_request, runtime)
    except Exception as error:
        print(error)
        if hasattr(error, 'message'):
            print(error.message)
        if hasattr(error, 'data') and error.data.get("Recommend"):
            print(error.data.get("Recommend"))
        UtilClient.assert_as_string(str(error))

if __name__ == '__main__':
    Sample.main(sys.argv[1:])

# python scripts/auto-ss-config/aliyun_elastic_ip_manager.py unbind --allocation_id eip-j6c2olusa7jk9l42i1aaa
# python scripts/auto-ss-config/aliyun_elastic_ip_manager.py bind --allocation_id eip-j6c7mhenamvy6zao3haaa
# python scripts/auto-ss-config/aliyun_elastic_ip_manager.py release --allocation_id "eip-j6c2olusa7jk9l42i"
# python scripts/auto-ss-config/aliyun_elastic_ip_manager.py describe

```