aws-devops-cicd使用aws云服务CodeBuild和CodeDeploy搭建cicd持续集成与持续交付pipeline流水线-吕蕴偲Kevin-LYU

欢迎关注我的b站账号 space.bilibili.com/190926209

YouTube账号 www.youtube.com/@LYU_KEVIN

获取更多aws及云计算视频教程

本文属于机器翻译版本。若本译文内容与英语原文存在差异，则一律以英文原文为准。

使用aws云服务CodeBuild和CodeDeploy搭建cicd持续集成与持续交付pipeline流水线(架构图)

DevOps 管道示例 – AWS CodePipeline

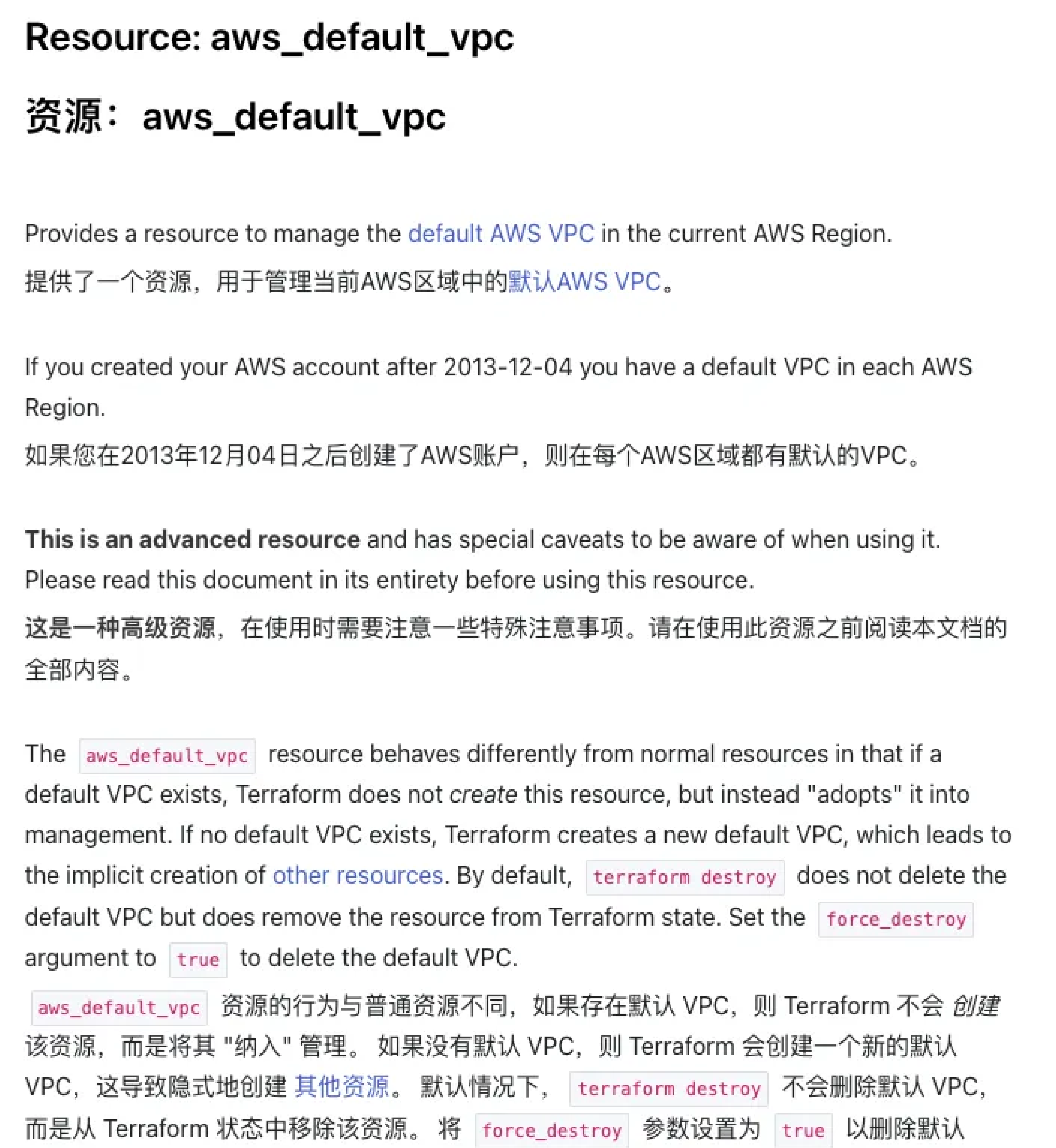
https://docs.aws.amazon.com/zh_cn/codepipeline/latest/userguide/concepts-devops-example.html

适用于 CodeBuild 的构建规范参考 – AWS CodeBuild:

https://docs.aws.amazon.com/zh_cn/codebuild/latest/userguide/build-spec-ref.html#build-spec-ref-syntax

aws_default_vpc

https://registry.terraform.io/providers/hashicorp/aws/latest/docs/resources/default_vpc



Resource: aws_default_vpc

资源：aws_default_vpc

Provides a resource to manage the default AWS VPC in the current AWS Region.

提供了一个资源，用于管理当前AWS区域中的默认AWS VPC。

If you created your AWS account after 2013-12-04 you have a default VPC in each AWS Region.

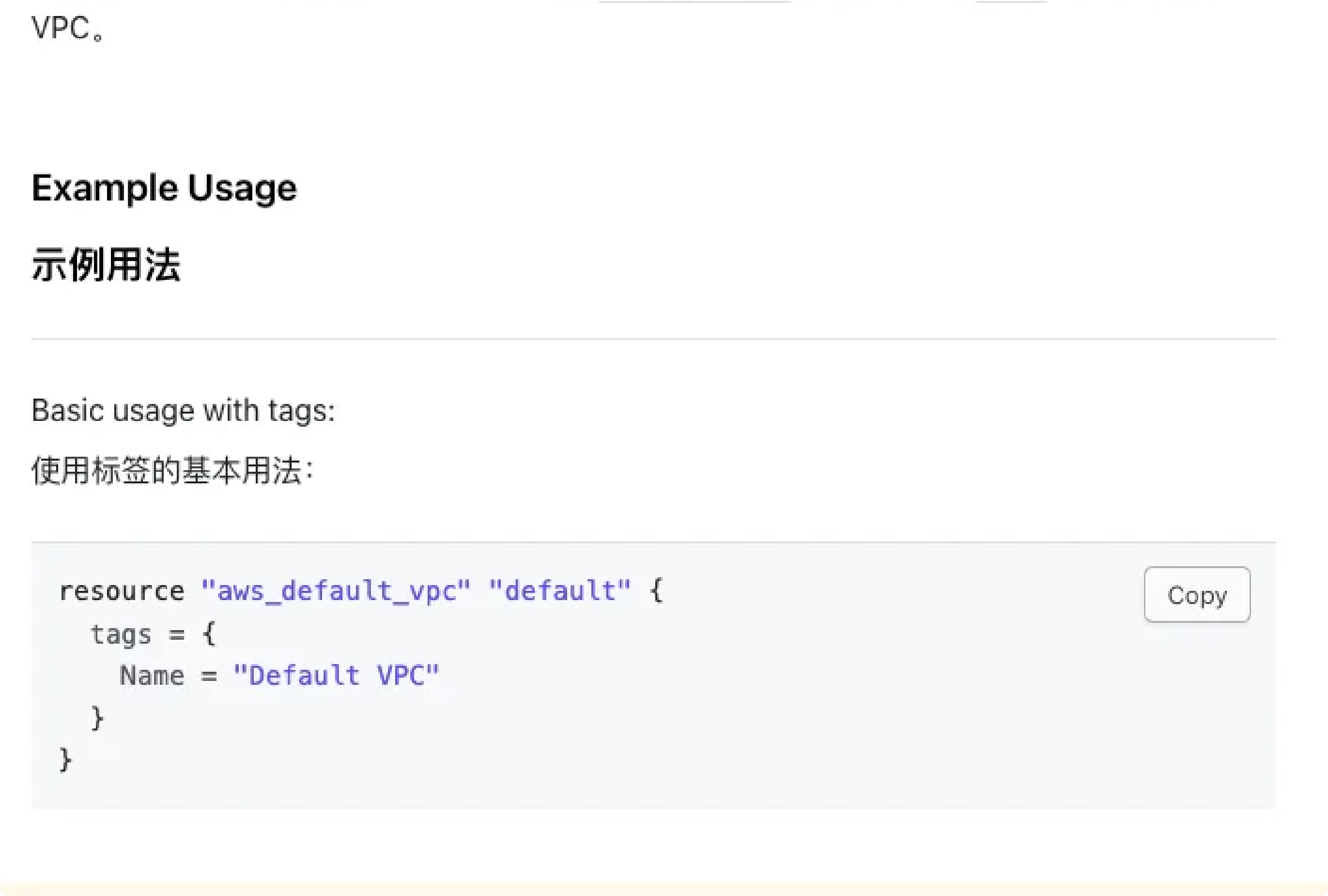
如果您在2013年12月04日之后创建了AWS账户，则在每个AWS区域都有默认的VPC。

This is an advanced resource and has special caveats to be aware of when using it. Please read this document in its entirety before using this resource.

这是一种高级资源，在使用时需要注意一些特殊注意事项。请在使用此资源之前阅读本文档的全部内容。

The `aws_default_vpc` resource behaves differently from normal resources in that if a default VPC exists, Terraform does not *create* this resource, but instead "adopts" it into management. If no default VPC exists, Terraform creates a new default VPC, which leads to the implicit creation of other resources. By default, `terraform destroy` does not delete the default VPC but does remove the resource from Terraform state. Set the `force_destroy` argument to `true` to delete the default VPC.

`aws_default_vpc` 资源的行为与普通资源不同，如果存在默认 VPC，则 Terraform 不会 *创建* 该资源，而是将其 "纳入" 管理。 如果没有默认 VPC，则 Terraform 会创建一个新的默认 VPC，这导致隐式地创建 其他资源。 默认情况下， `terraform destroy` 不会删除默认 VPC，而是从 Terraform 状态中移除该资源。 将 `force_destroy` 参数设置为 `true` 以删除默认



VPC。

Example Usage

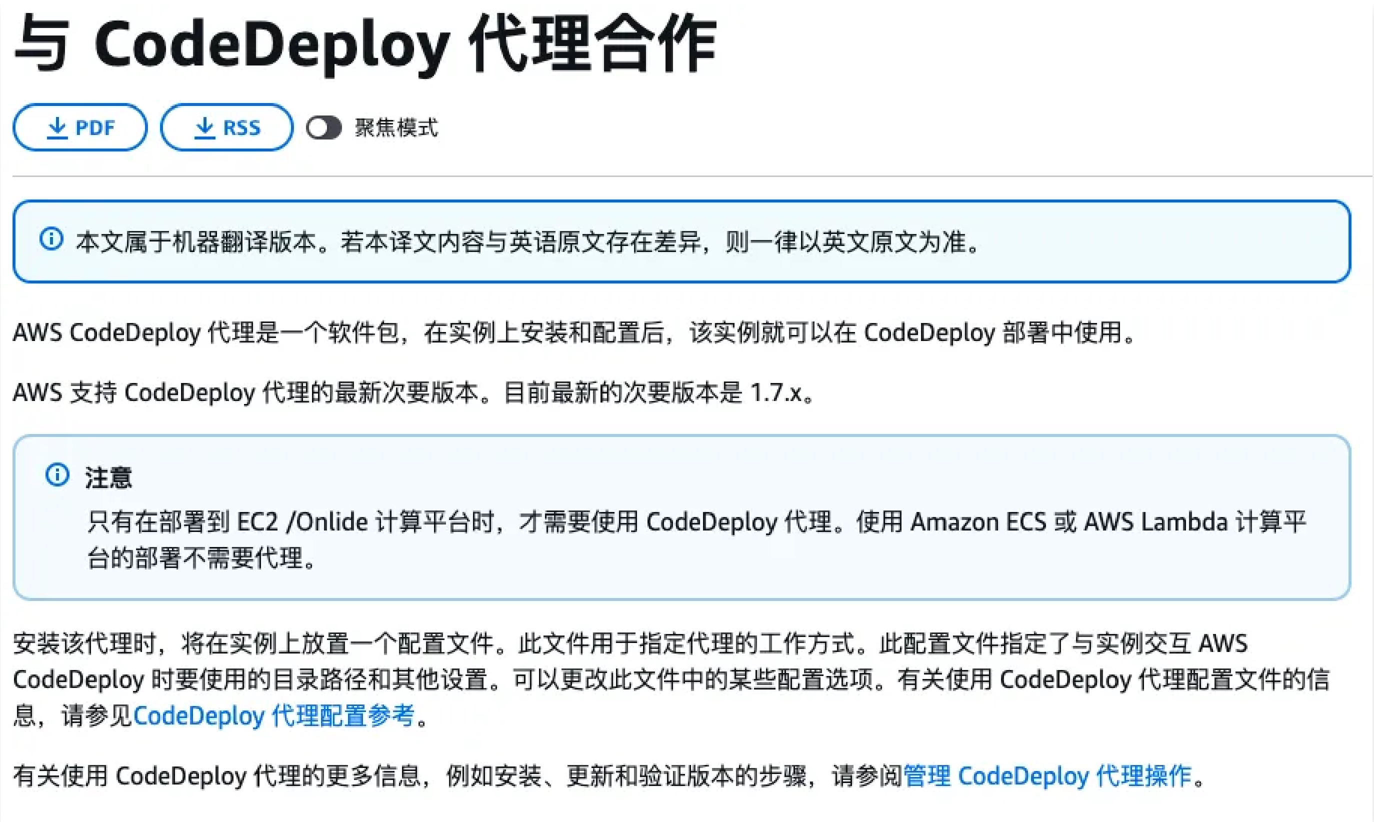
示例用法

Basic usage with tags:

使用标签的基本用法:

```
resource "aws_default_vpc" "default" {
  tags = {
    Name = "Default VPC"
  }
}
```

安装CodeDeploy agent



与 CodeDeploy 代理合作

本文属于机器翻译版本。若本译文内容与英语原文存在差异，则一律以英文原文为准。

AWS CodeDeploy 代理是一个软件包，在实例上安装和配置后，该实例就可以在 CodeDeploy 部署中使用。

AWS 支持 CodeDeploy 代理的最新次要版本。目前最新的次要版本是 1.7.x。

注意

只有在部署到 EC2 /Onlide 计算平台时，才需要使用 CodeDeploy 代理。使用 Amazon ECS 或 AWS Lambda 计算平台的部署不需要代理。

安装该代理时，将在实例上放置一个配置文件。此文件用于指定代理的工作方式。此配置文件指定了与实例交互 AWS CodeDeploy 时要使用的目录路径和其他设置。可以更改此文件中的某些配置选项。有关使用 CodeDeploy 代理配置文件的信息，请参见CodeDeploy 代理配置参考。

有关使用 CodeDeploy 代理的更多信息，例如安装、更新和验证版本的步骤，请参阅管理 CodeDeploy 代理操作。

https://docs.aws.amazon.com/zh_cn/codedeploy/latest/userguide/codedeploy-agent-operations-install-linux.html

awc configure

aws configure set aws_access_key_id $AWS_ACCESS_KEY_ID

aws configure set aws_secret_access_key $AWS_SECRET_ACCESS_KEY

aws configure set region $AWS_REGION

使用安装 CodeDeploy 代理

https://docs.aws.amazon.com/zh_cn/codedeploy/latest/userguide/codedeploy-agent-operations-install-ssm.html

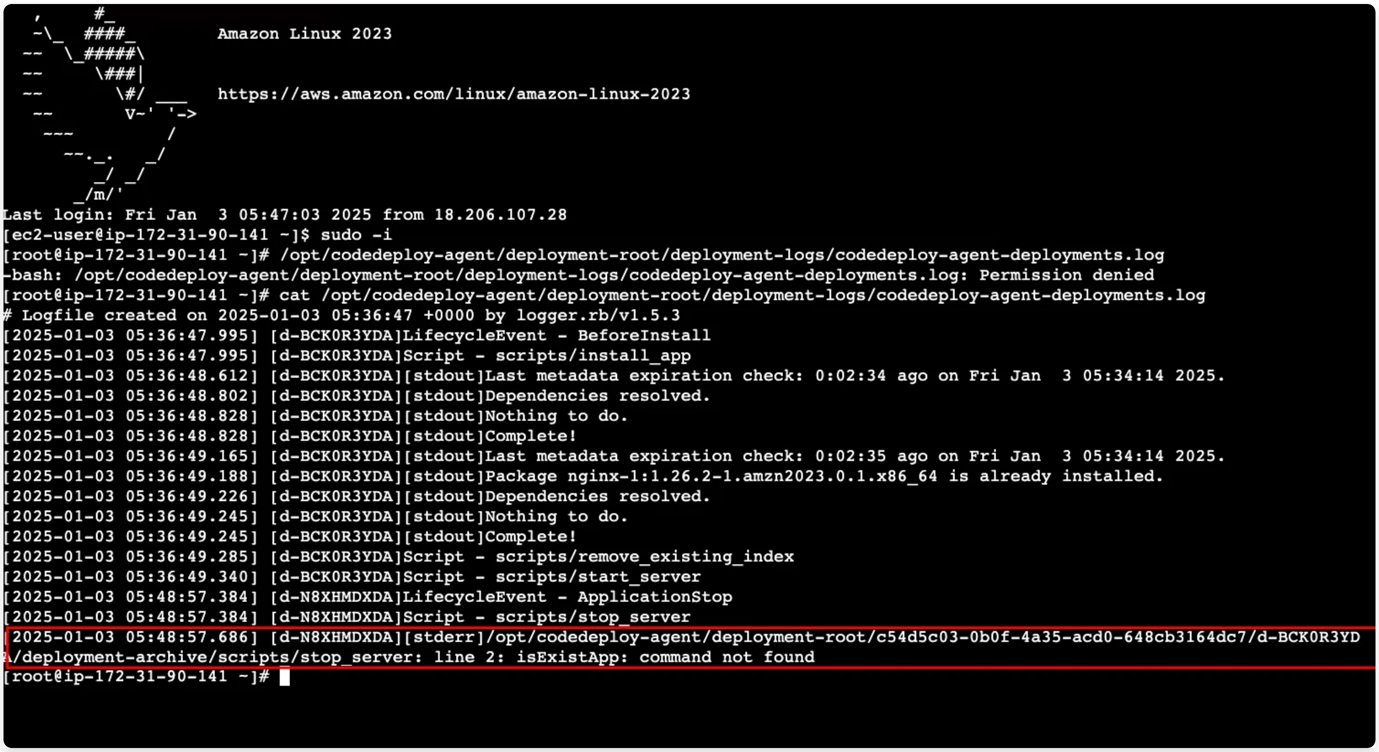
安装或更新 Distributor 软件包 – AWS Systems Manager

https://docs.aws.amazon.com/zh_cn/systems-manager/latest/userguide/distributor-working-with-packages-deploy.html#distributor-deploy-sm-pkg-console

查看 CodeDeploy EC2/本地部署的日志数据 – AWS CodeDeploy

https://docs.aws.amazon.com/zh_cn/codedeploy/latest/userguide/deployments-view-logs.html

如果使用cicd直接同一个流水线部署，可能会出现如下报错，原因是没有读取到source的repo仓库。



```
   ,     #_
   ~\_  ####_        Amazon Linux 2023
  ~~  \_#####\
  ~~     \###|
  ~~       \#/ ___   https://aws.amazon.com/linux/amazon-linux-2023
   ~~       V~' '->
    ~~~         /
      ~~._.   _/
         _/ _/
       _/m/'
Last login: Fri Jan  3 05:47:03 2025 from 18.206.107.28
[ec2-user@ip-172-31-90-141 ~]$ sudo -i
[root@ip-172-31-90-141 ~]# /opt/codedeploy-agent/deployment-root/deployment-logs/codedeploy-agent-deployments.log
-bash: /opt/codedeploy-agent/deployment-root/deployment-logs/codedeploy-agent-deployments.log: Permission denied
[root@ip-172-31-90-141 ~]# cat /opt/codedeploy-agent/deployment-root/deployment-logs/codedeploy-agent-deployments.log
# Logfile created on 2025-01-03 05:36:47 +0000 by logger.rb/v1.5.3
[2025-01-03 05:36:47.995] [d-BCK0R3YDA]LifecycleEvent - BeforeInstall
[2025-01-03 05:36:47.995] [d-BCK0R3YDA]Script - scripts/install_app
[2025-01-03 05:36:48.612] [d-BCK0R3YDA][stdout]Last metadata expiration check: 0:02:34 ago on Fri Jan  3 05:34:14 2025.
[2025-01-03 05:36:48.802] [d-BCK0R3YDA][stdout]Dependencies resolved.
[2025-01-03 05:36:48.828] [d-BCK0R3YDA][stdout]Nothing to do.
[2025-01-03 05:36:48.828] [d-BCK0R3YDA][stdout]Complete!
[2025-01-03 05:36:49.165] [d-BCK0R3YDA][stdout]Last metadata expiration check: 0:02:35 ago on Fri Jan  3 05:34:14 2025.
[2025-01-03 05:36:49.188] [d-BCK0R3YDA][stdout]Package nginx-1:1.26.2-1.amzn2023.0.1.x86_64 is already installed.
[2025-01-03 05:36:49.226] [d-BCK0R3YDA][stdout]Dependencies resolved.
[2025-01-03 05:36:49.245] [d-BCK0R3YDA][stdout]Nothing to do.
[2025-01-03 05:36:49.245] [d-BCK0R3YDA][stdout]Complete!
[2025-01-03 05:36:49.285] [d-BCK0R3YDA]Script - scripts/remove_existing_index
[2025-01-03 05:36:49.340] [d-BCK0R3YDA]Script - scripts/start_server
[2025-01-03 05:48:57.384] [d-N8XHMDXDA]LifecycleEvent - ApplicationStop
[2025-01-03 05:48:57.384] [d-N8XHMDXDA]Script - scripts/stop_server
[2025-01-03 05:48:57.686] [d-N8XHMDXDA][stderr]/opt/codedeploy-agent/deployment-root/c54d5c03-0b0f-4a35-acd0-648cb3164dc7/d-BCK0R3YD
A/deployment-archive/scripts/stop_server: line 2: isExistApp: command not found
[root@ip-172-31-90-141 ~]#
```

在预留容量实例集上运行构建 – AWS CodeBuild

https://docs.aws.amazon.com/zh_cn/codebuild/latest/userguide/fleets.html#fleets.share

CodePipeline 管道结构参考 – AWS CodePipeline

https://docs.aws.amazon.com/zh_cn/codepipeline/latest/userguide/reference-pipeline-structure.html?icmpid=docs_acp_console#input-output-artifacts

每个动作类型有效的输入和输出工件 – AWS CodePipeline

https://docs.aws.amazon.com/codepipeline/latest/userguide/reference-action-artifacts.html

教程：使用管道级变量 – AWS CodePipeline

https://docs.aws.amazon.com/zh_cn/codepipeline/latest/userguide/tutorials-pipeline-variables.html

CodeDeploy 资源包参考 – AWS CodeDeploy

https://docs.aws.amazon.com/zh_cn/codedeploy/latest/userguide/resource-kit.html#resource-kit-bucket-names

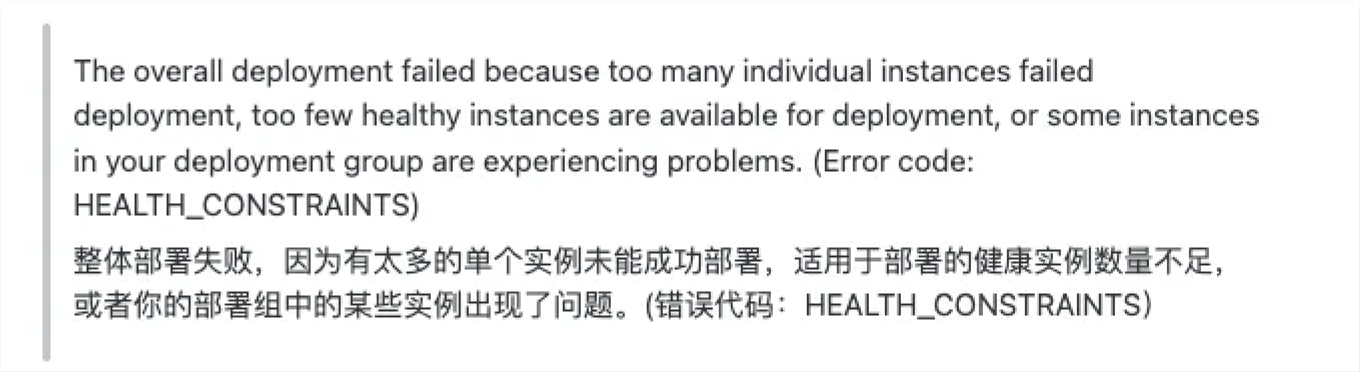
The overall deployment failed because too many individual instances failed deployment – Stack Overflow error整体部署失败，因为太多的单个实例失败部署

如果是在启动ec2后附加的iam策略，需要重启ec2，或者agent，否则agent认不到iam角色，自然就没有权限访问提示error

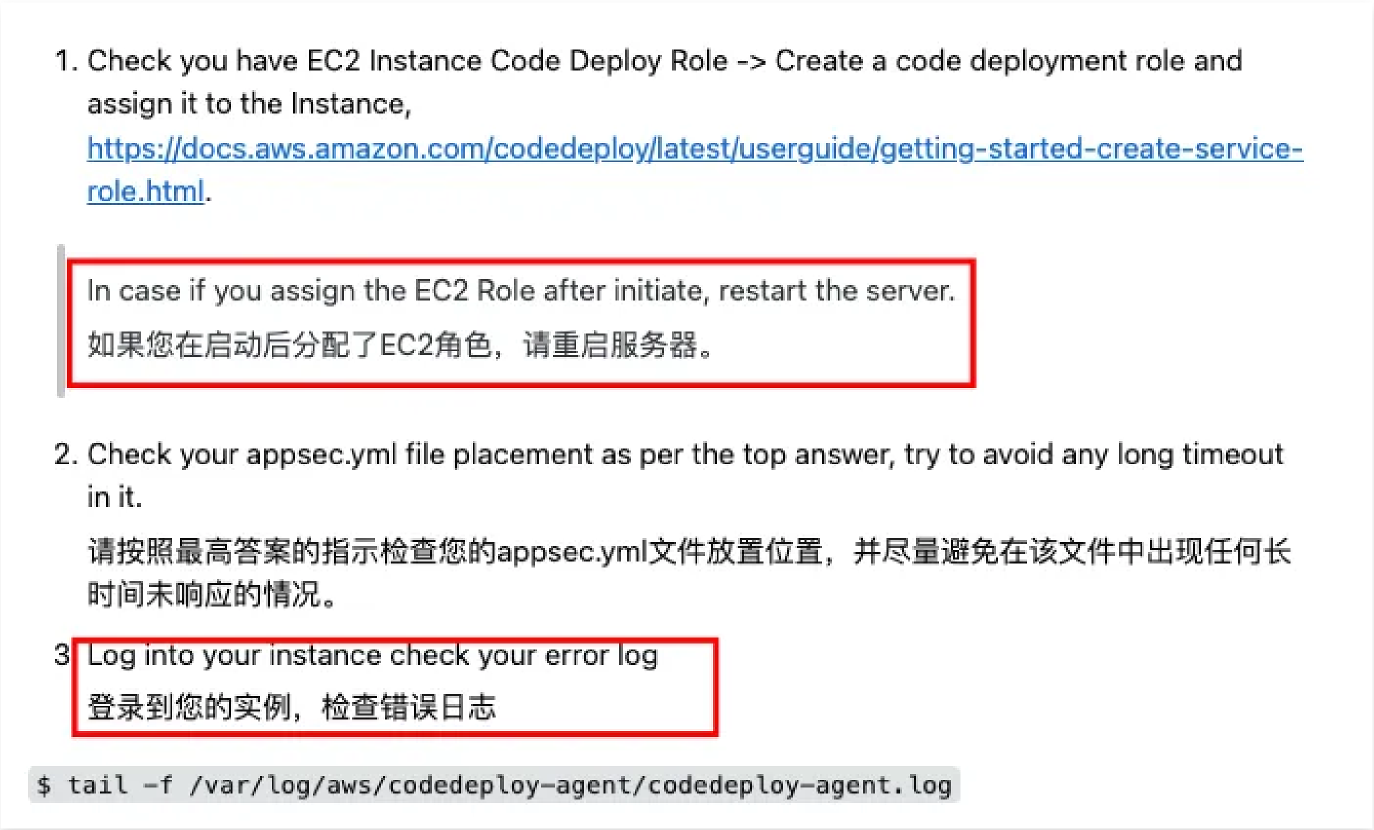
https://stackoverflow.com/questions/38195823/error-the-overall-deployment-failed-because-too-many-individual-instances-faile

agent错误日志，上报服务器端超时或失败可以看这个日志

```
tail -n 500 /var/log/aws/codedeploy-agent/codedeploy-agent.log
```



> The overall deployment failed because too many individual instances failed deployment, too few healthy instances are available for deployment, or some instances in your deployment group are experiencing problems. (Error code: HEALTH_CONSTRAINTS)
>
> 整体部署失败，因为有太多的单个实例未能成功部署，适用于部署的健康实例数量不足，或者你的部署组中的某些实例出现了问题。(错误代码：HEALTH_CONSTRAINTS)



1. Check you have EC2 Instance Code Deploy Role -> Create a code deployment role and assign it to the Instance, https://docs.aws.amazon.com/codedeploy/latest/userguide/getting-started-create-service-role.html.

   > In case if you assign the EC2 Role after initiate, restart the server.
   >
   > 如果您在启动后分配了EC2角色，请重启服务器。

2. Check your appsec.yml file placement as per the top answer, try to avoid any long timeout in it.

   请按照最高答案的指示检查您的appsec.yml文件放置位置，并尽量避免在该文件中出现任何长时间未响应的情况。

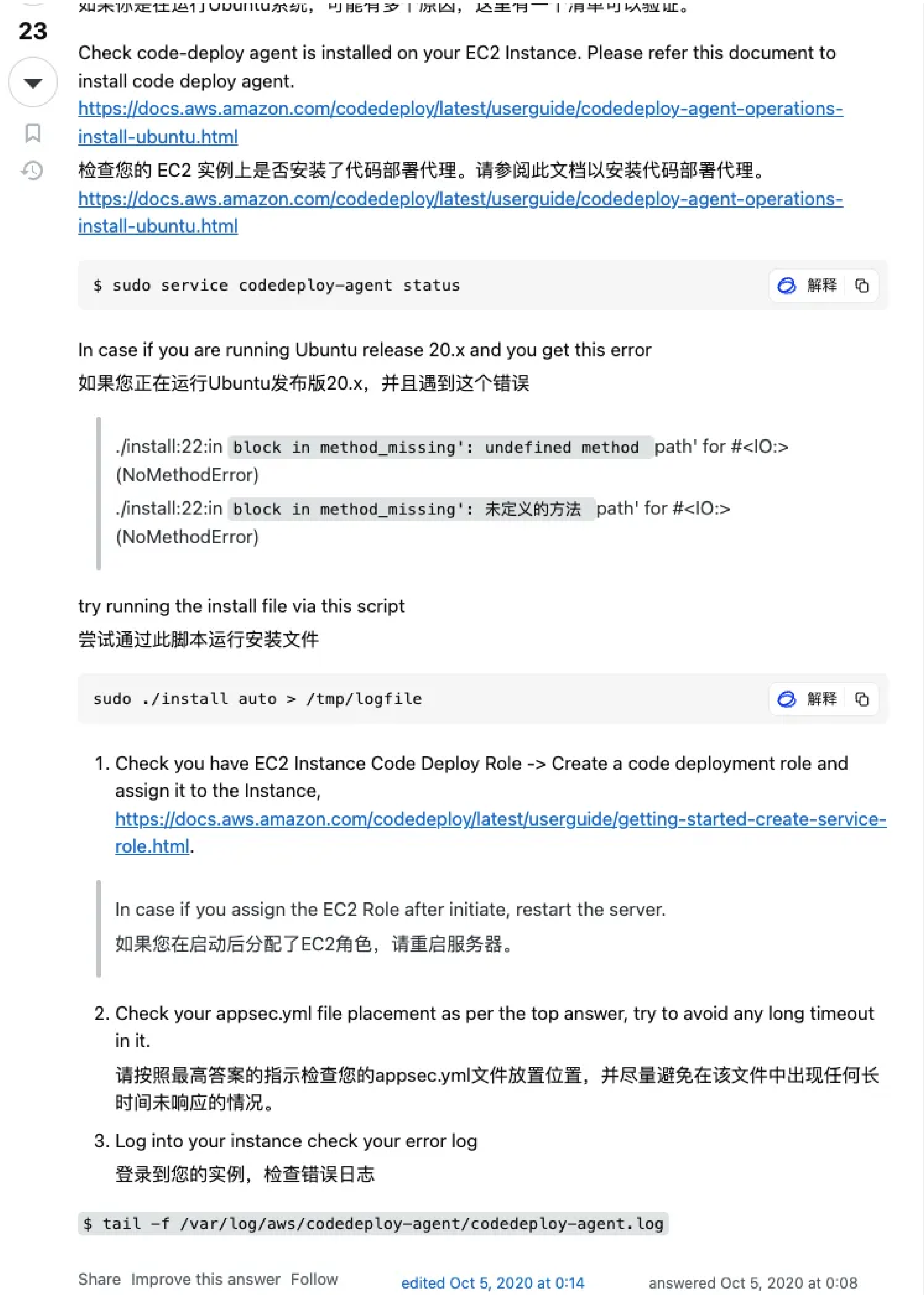
3. Log into your instance check your error log

   登录到您的实例，检查错误日志

```
$ tail -f /var/log/aws/codedeploy-agent/codedeploy-agent.log
```

If you are running on Ubuntu there might be plenty of reasons, here is a checklist can verify

如果你是在运行Ubuntu系统，可能有多个原因，这里有一个清单可以验证



如果你是在运行Ubuntu系统，可能有多个原因，这里有一个清单可以验证。

Check code-deploy agent is installed on your EC2 Instance. Please refer this document to install code deploy agent. https://docs.aws.amazon.com/codedeploy/latest/userguide/codedeploy-agent-operations-install-ubuntu.html

检查您的 EC2 实例上是否安装了代码部署代理。请参阅此文档以安装代码部署代理。https://docs.aws.amazon.com/codedeploy/latest/userguide/codedeploy-agent-operations-install-ubuntu.html

```
$ sudo service codedeploy-agent status
```

In case if you are running Ubuntu release 20.x and you get this error

如果您正在运行Ubuntu发布版20.x，并且遇到这个错误

> ./install:22:in `block in method_missing': undefined method` path' for #<IO:> (NoMethodError)
>
> ./install:22:in `block in method_missing': 未定义的方法` path' for #<IO:> (NoMethodError)

try running the install file via this script

尝试通过此脚本运行安装文件

```
sudo ./install auto > /tmp/logfile
```

1. Check you have EC2 Instance Code Deploy Role -> Create a code deployment role and assign it to the Instance, https://docs.aws.amazon.com/codedeploy/latest/userguide/getting-started-create-service-role.html.

   > In case if you assign the EC2 Role after initiate, restart the server.
   >
   > 如果您在启动后分配了EC2角色，请重启服务器。

2. Check your appsec.yml file placement as per the top answer, try to avoid any long timeout in it.

   请按照最高答案的指示检查您的appsec.yml文件放置位置，并尽量避免在该文件中出现任何长时间未响应的情况。

3. Log into your instance check your error log

   登录到您的实例，检查错误日志

`$ tail -f /var/log/aws/codedeploy-agent/codedeploy-agent.log`

Share Improve this answer Follow

edited Oct 5, 2020 at 0:14

answered Oct 5, 2020 at 0:08

分享　改进此答案

编辑于2020 年 10 月 5 日 00:14

kishorekumaru
1,578 ●2 ●22 ●34

3 that `tail` command saved me, ty – d8aninja Mar 13, 2022 at 20:32
2022 年 3 月 13 日 20:32

I attached the role after the instance was launched. Did not know, I should reboot the server. Thanks! Working fine. – Shams Nahid Feb 8, 2024 at 10:59
2024年2月8日 10:59

Add a comment
添加评论