Mendix私有云平台运维

【腾讯文档】Mendix私有云平台运维操作 https://docs.qq.com/doc/DU1FlSkV2WXBXYWZM

## 通过terraform创建EKS环境

### 获取代码

# 获取代码

### 配置项目参数

参数配置

# variables.tf 基础值处理

aws\_region = "cn-northwest-1"

# 域名

domain\_name = ""

# EKS 名称前缀

eks\_cluster\_name\_prefix = ""

# 联系邮箱 可选

certificate\_expiration\_email = "qukaige@beijingwujian.com"

# S3桶名称

s3\_bucket\_name = ""

# 通过Mendix 私有平台获取，下面步骤中可见

namespace\_id = "89d2408a-2052-4b89-84fd-ee7e61fdff9b"

namespace\_secret = "04PeP8eF5kHQg61t"

# 环境名称

environments\_internal\_names = ["test"]

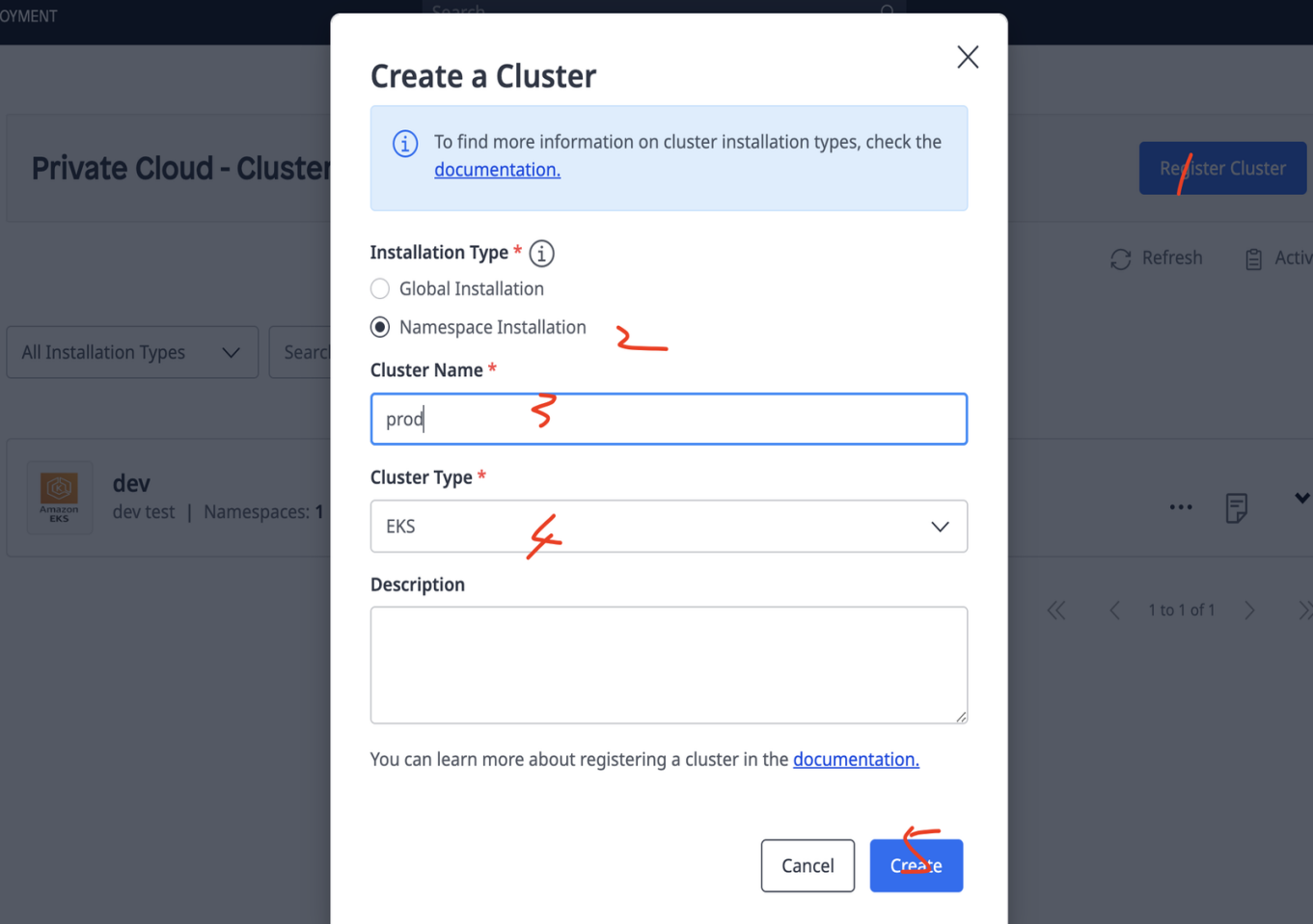
执行脚本

terraform init

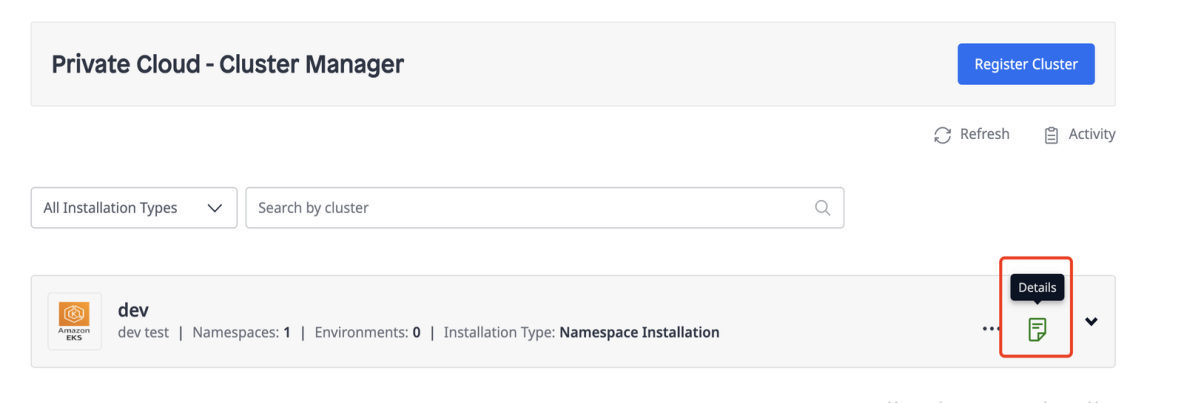
terraform apply

## Mendix私有平台连接EKS

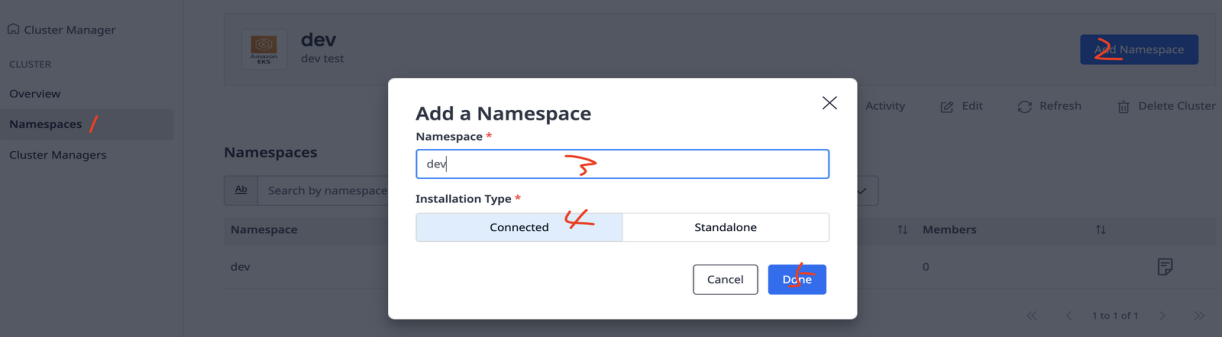
### 创建集群



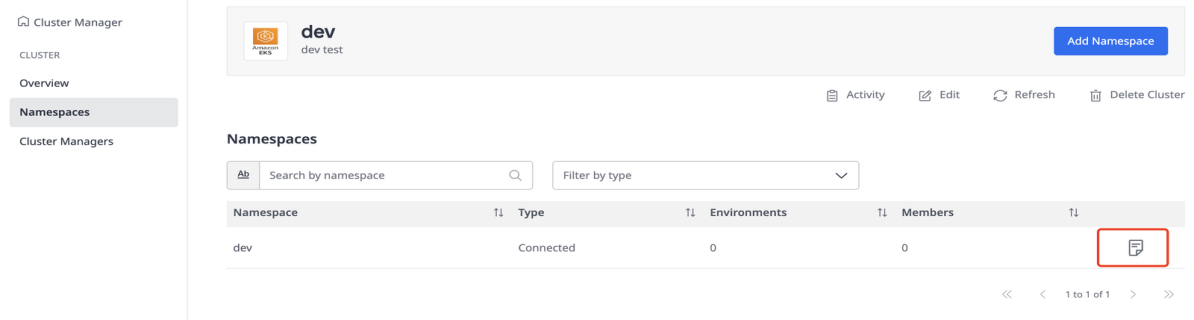
### 点击details进入集群



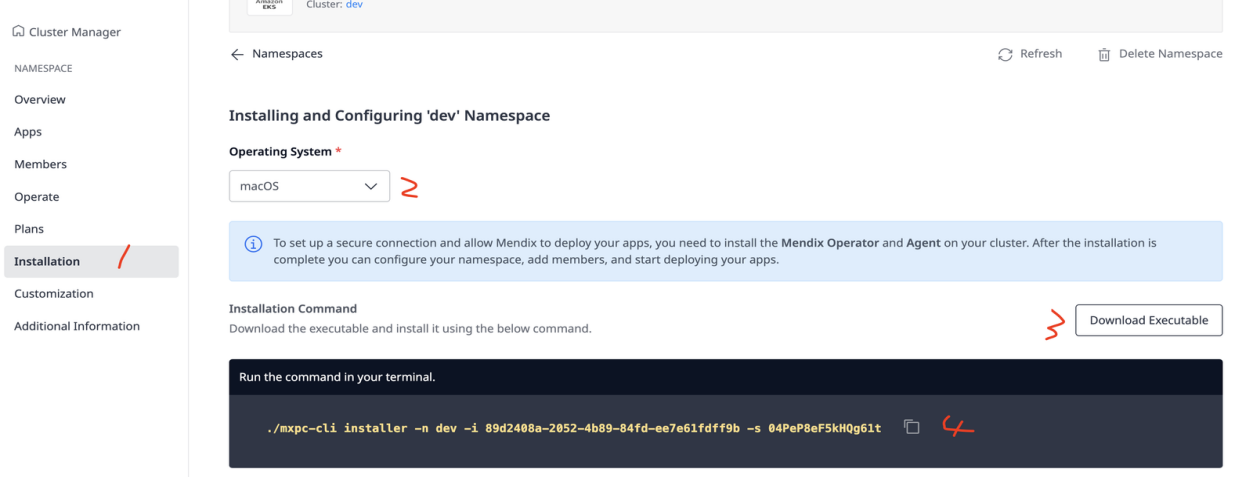
### 创建namespaces



### 进入namespace环境



### 工具下载



选择自己的系统环境， 执行4命令之前需要先3下载mxpc-cli

注意：这里的i 就是 namespace\_id , s 就是namespace\_secret

### 安装MendixOperator和Agent （1）

打开控制台 执行命令

./mxpc-cli installer -n dev -i xxx -s xxx

即可看到以下配置面板

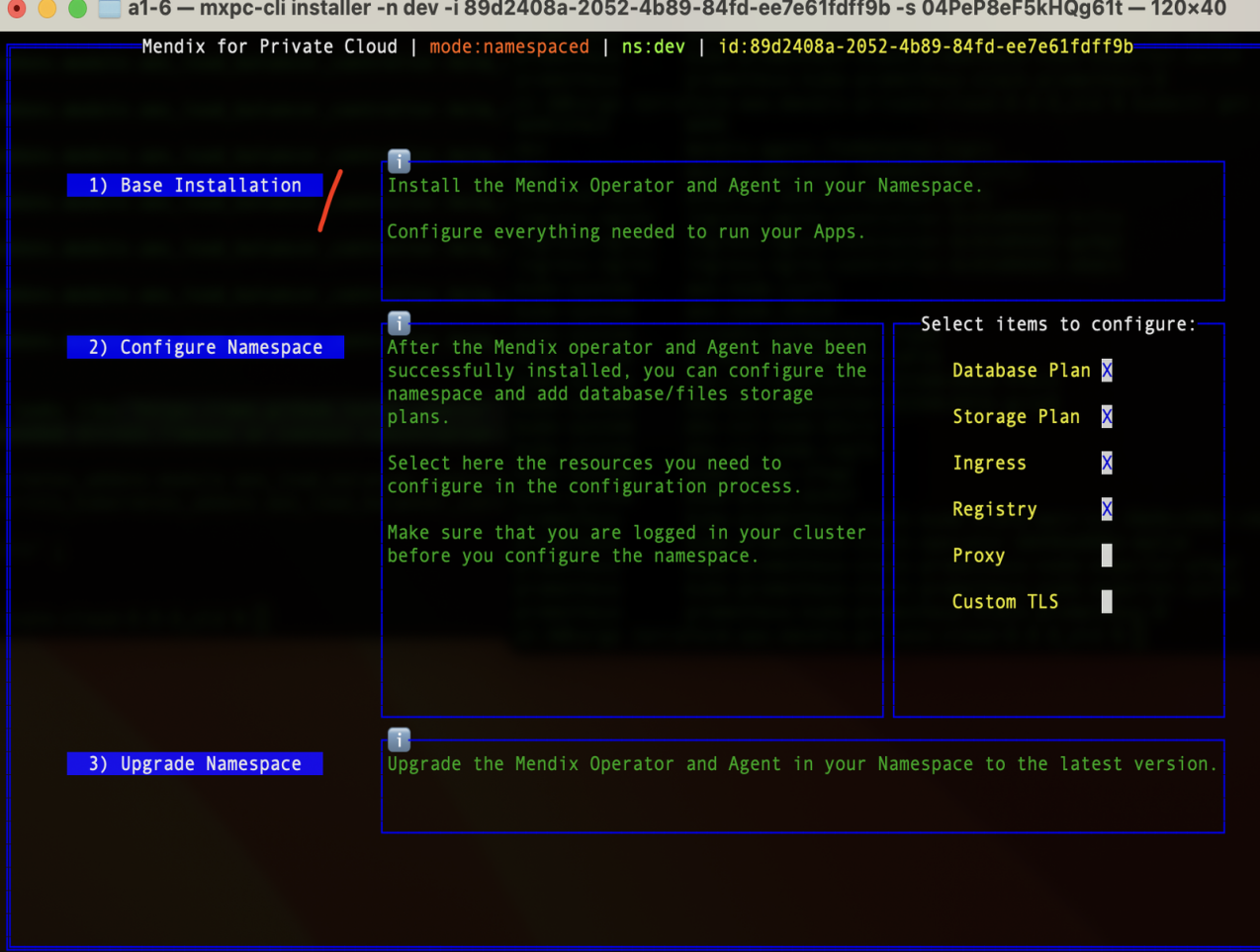
注意： 执行此命令前确保已连接EKS集群

# 连接集群命令

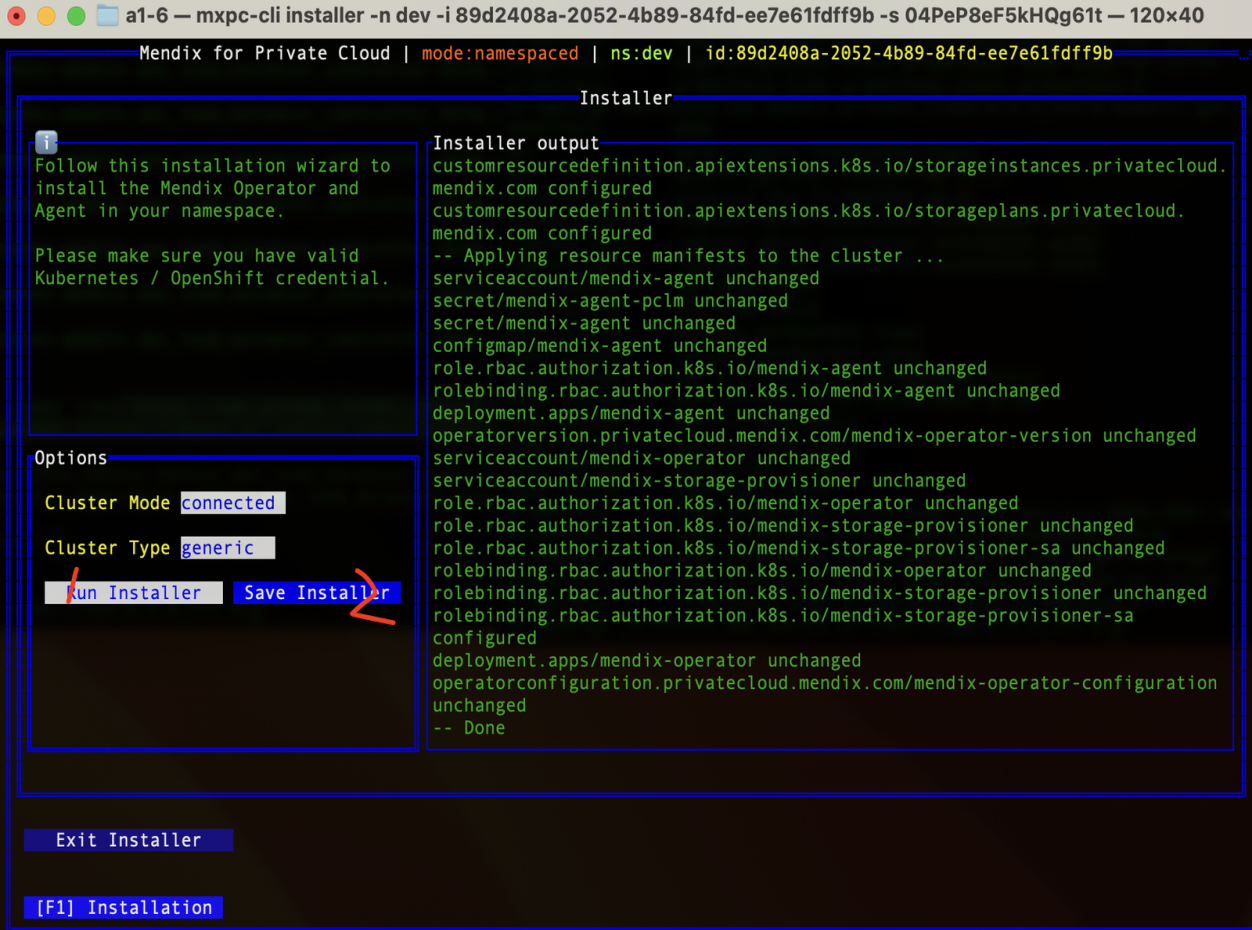
aws eks --region $(terraform output -raw region) update-kubeconfig --name $(terraform output -raw cluster\_name)

# 通过命令验证是否连接集群

kubectl cluster-info



点击Base Installation 跳转install 页面

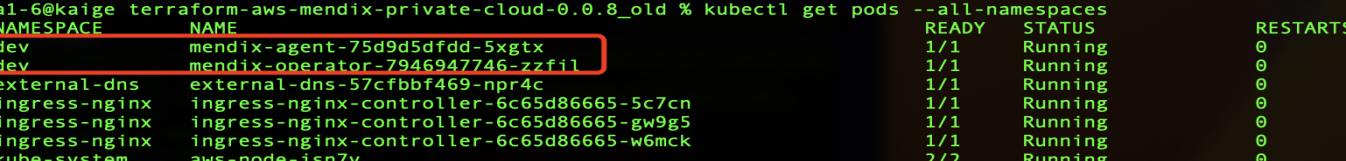


点击Run Installer 可以看到右侧显示Done **（第一步）已完成**

点击 Exit Installer 退出工具

查看已经安装的pod和namespace

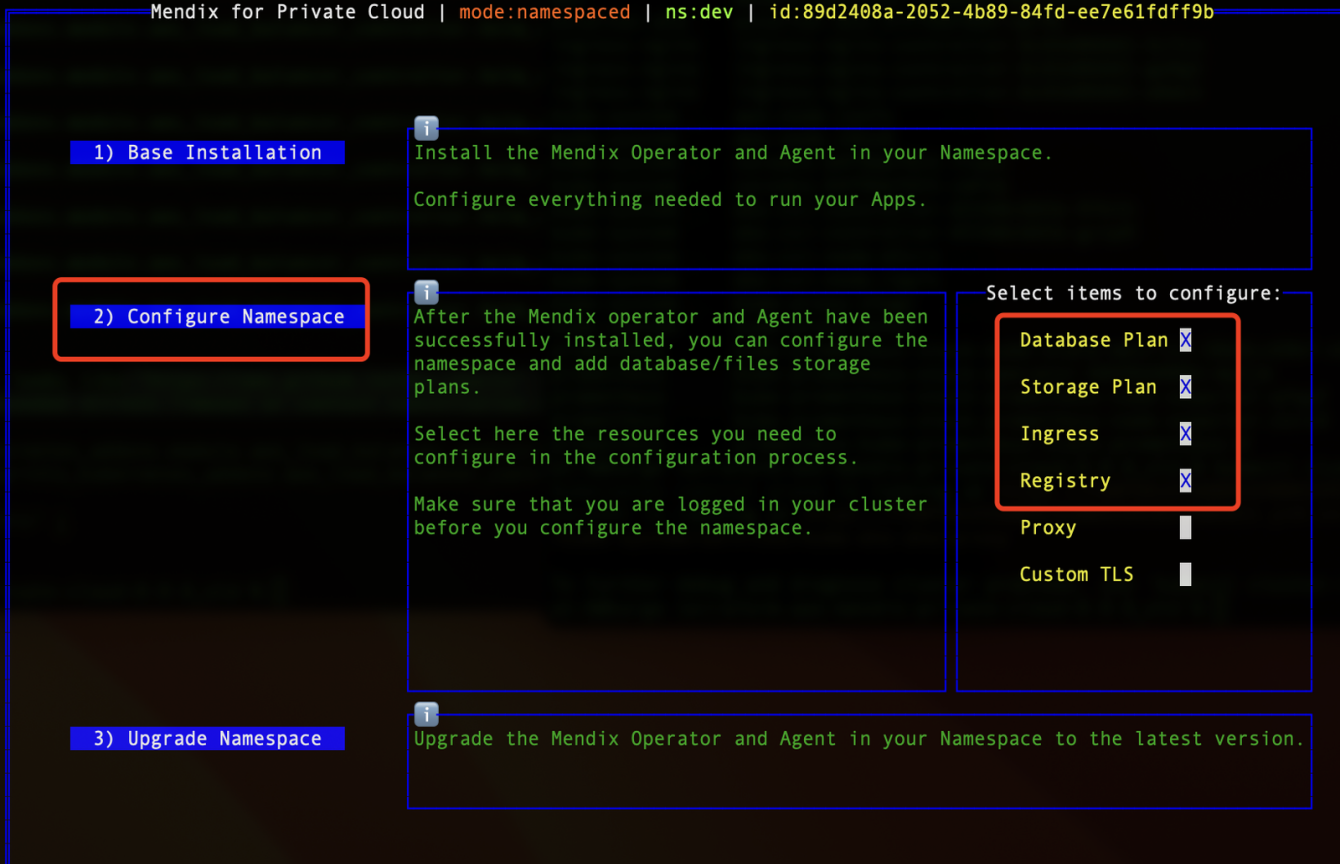
kubectl get pods --all-namespaces



### 配置数据库（2）

继续执行命令

./mxpc-cli installer -n dev -i xxx -s xxx



点击Configure Namespace

选择Database Plan， 填写自己的相关配置信息

Name 和 Database Type 保持一致即可

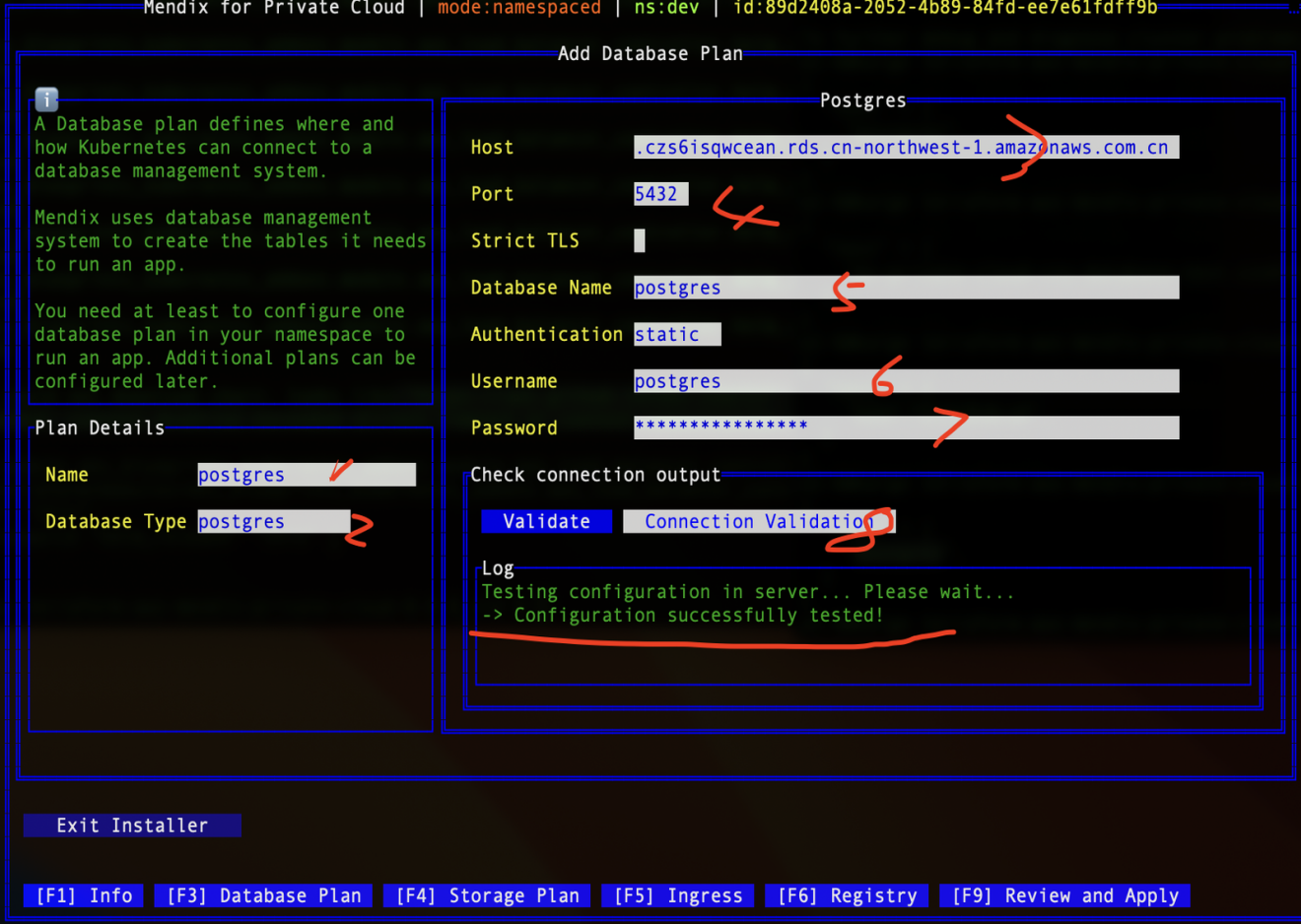
其它参数获取

terraform output database\_server\_address

terraform output database\_name

terraform output database\_username

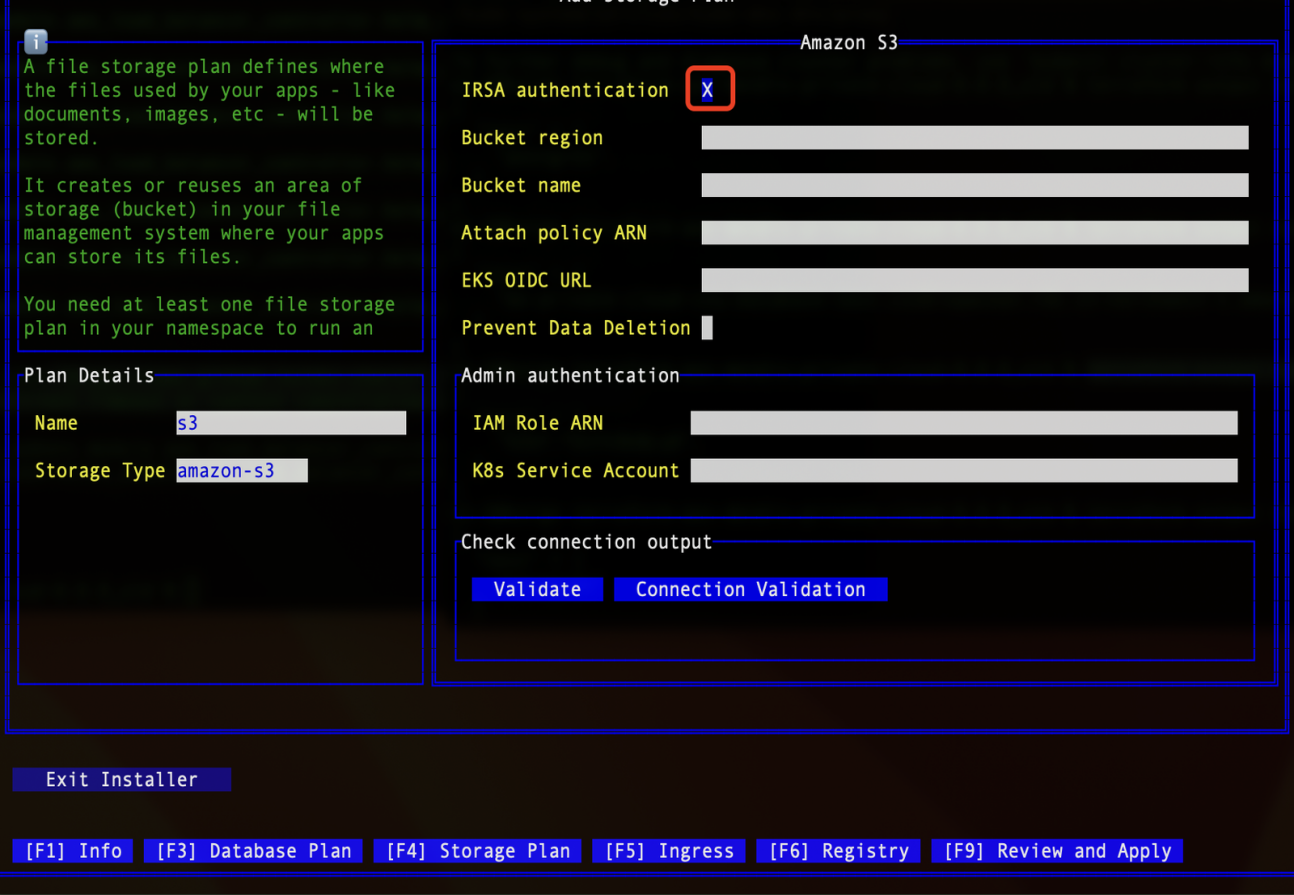
terraform output database\_password



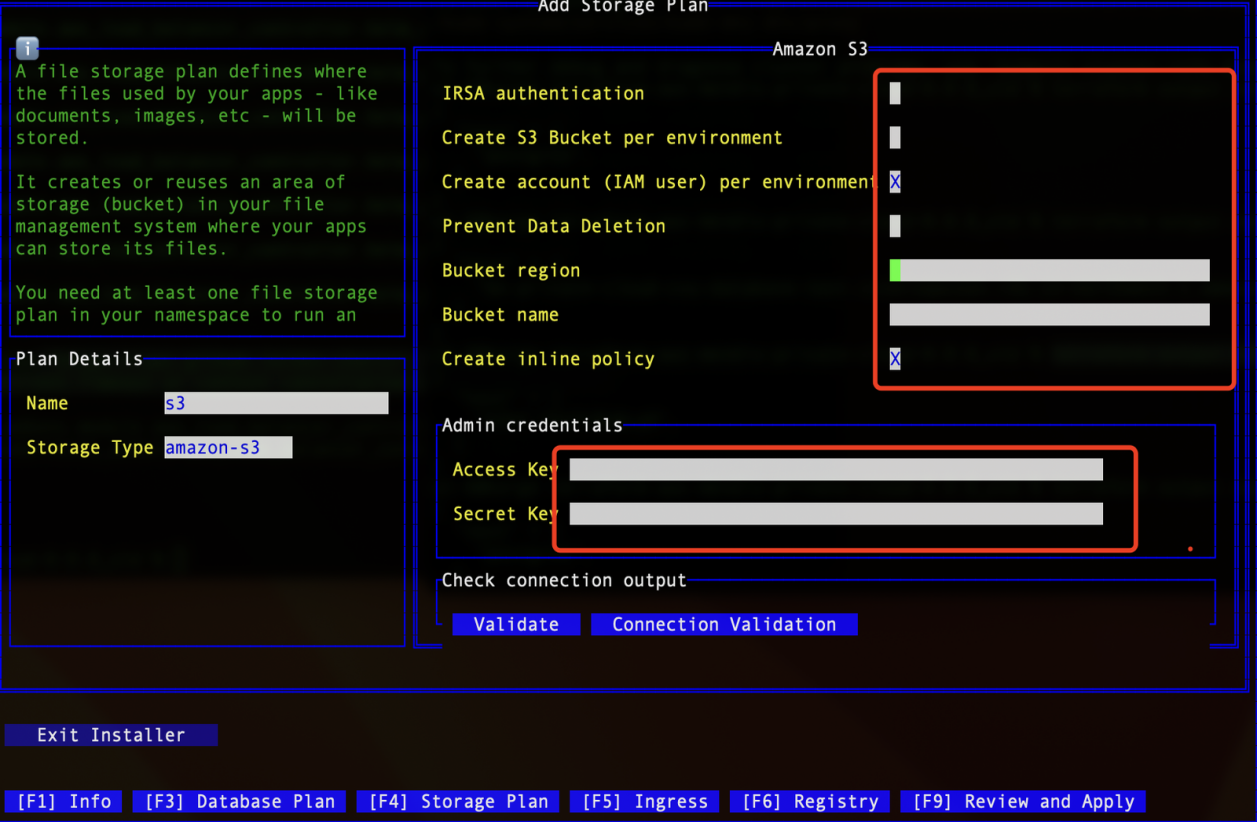
Connection Validation 验证配置是否正确

### 配置S3（3）

选择 F3 Storage Plan， 点击 IRSA authentication



保留这几个勾选



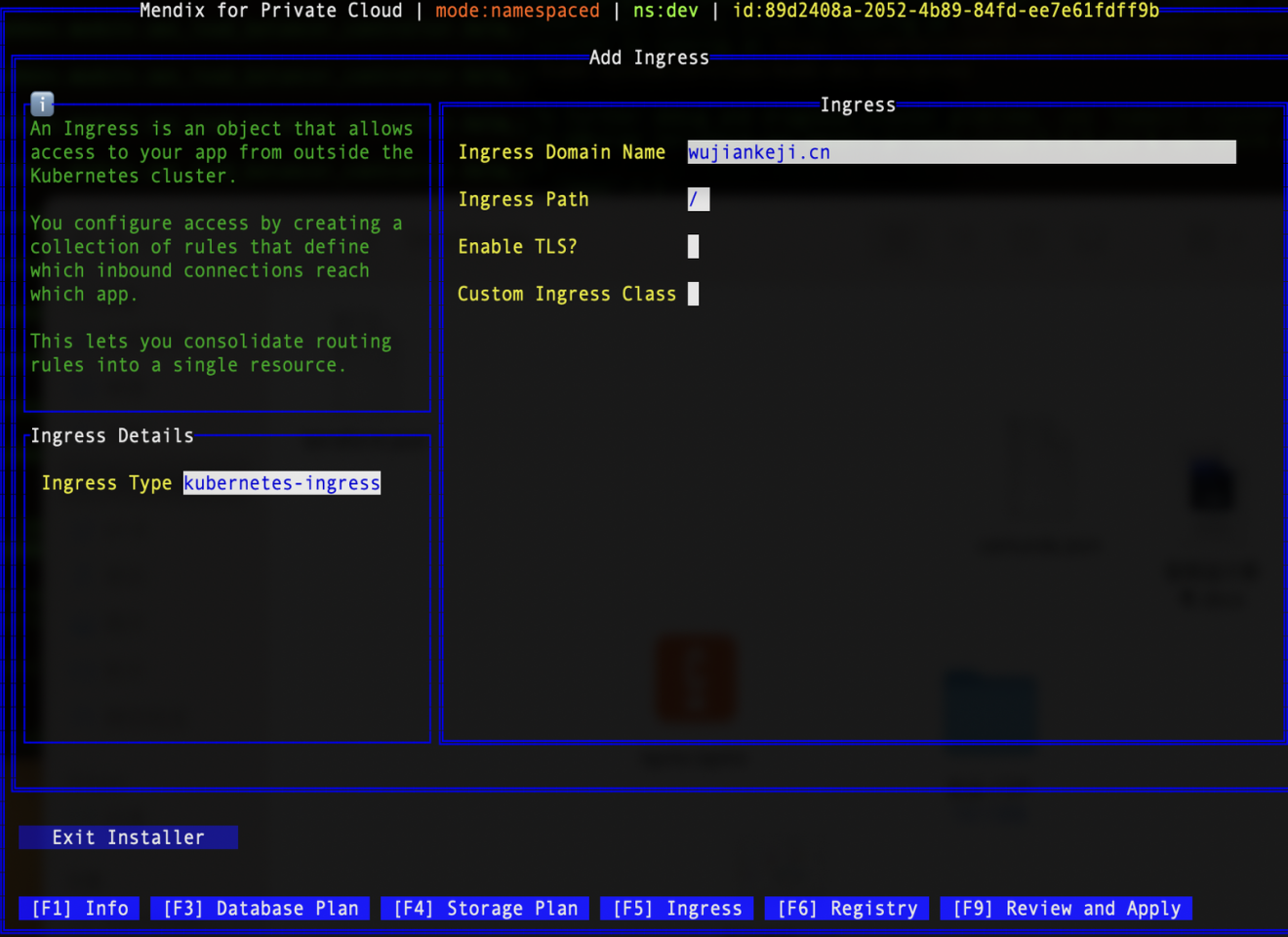
获取参数

terraform output region

bucket name terraform已经配置过

### 配置Ingress（4）

选择 F5 Ingress



### 配置Registry（5）

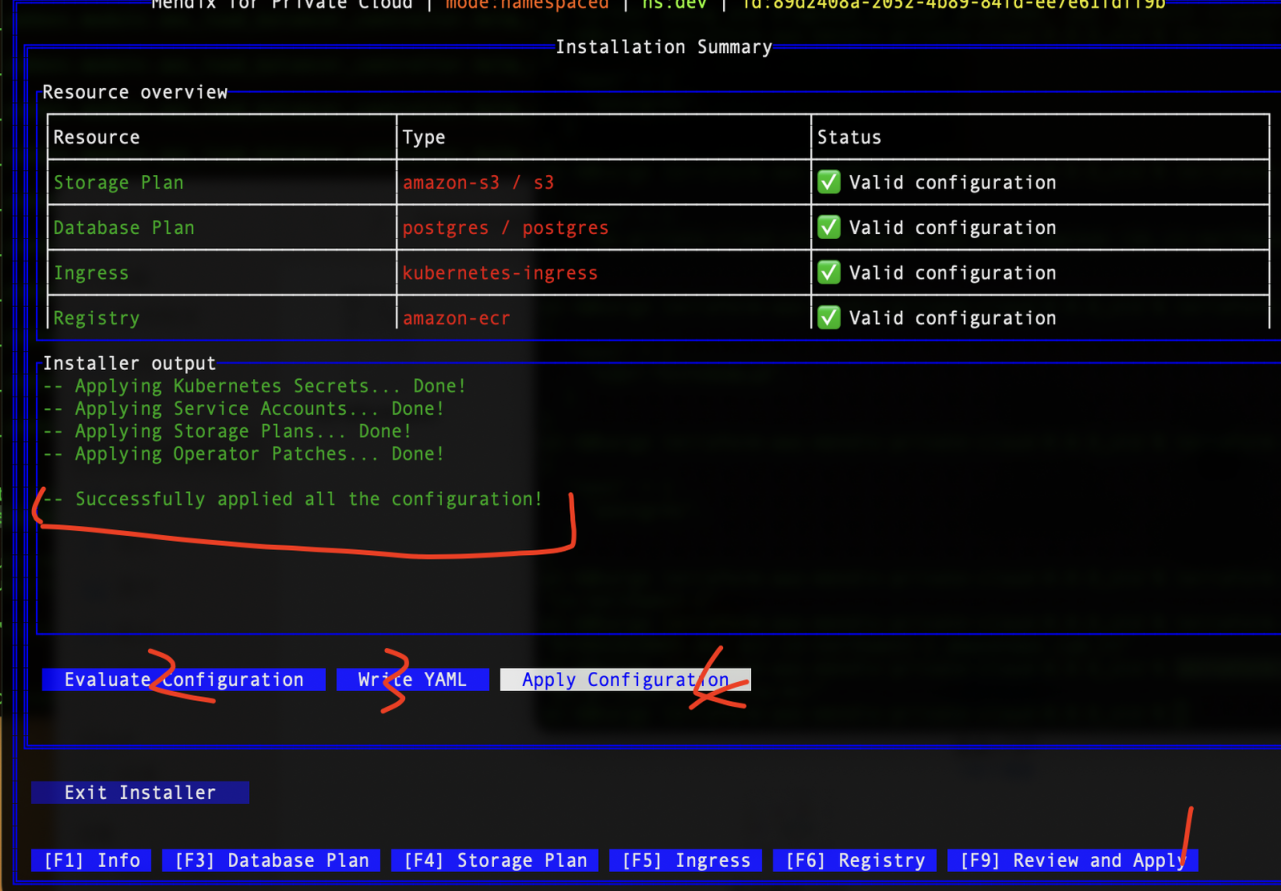
terraform output container\_registry\_url

terraform output container\_registry\_name



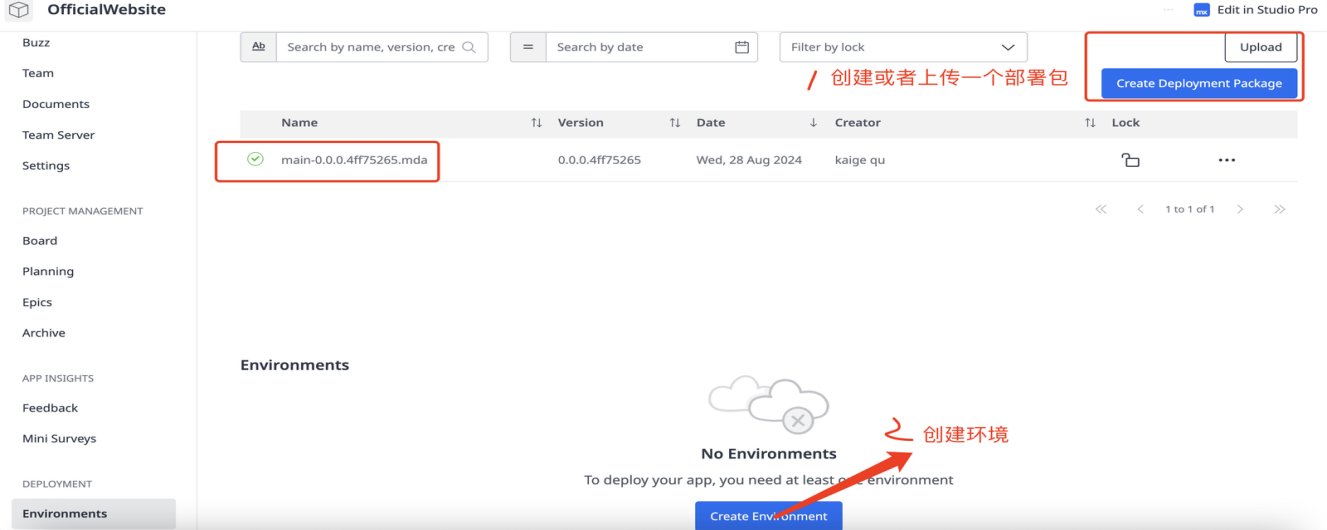
### 验证（6）

点击 Review And Apply

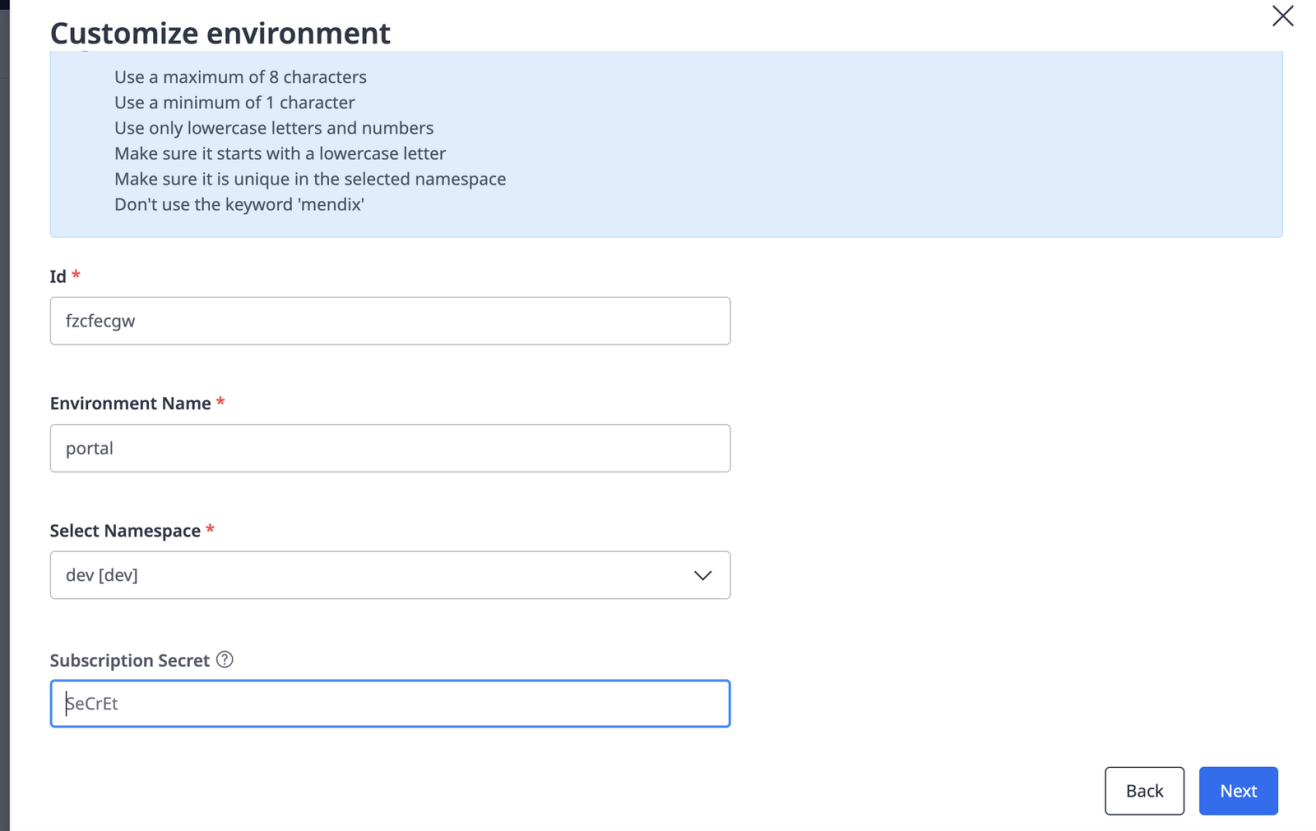


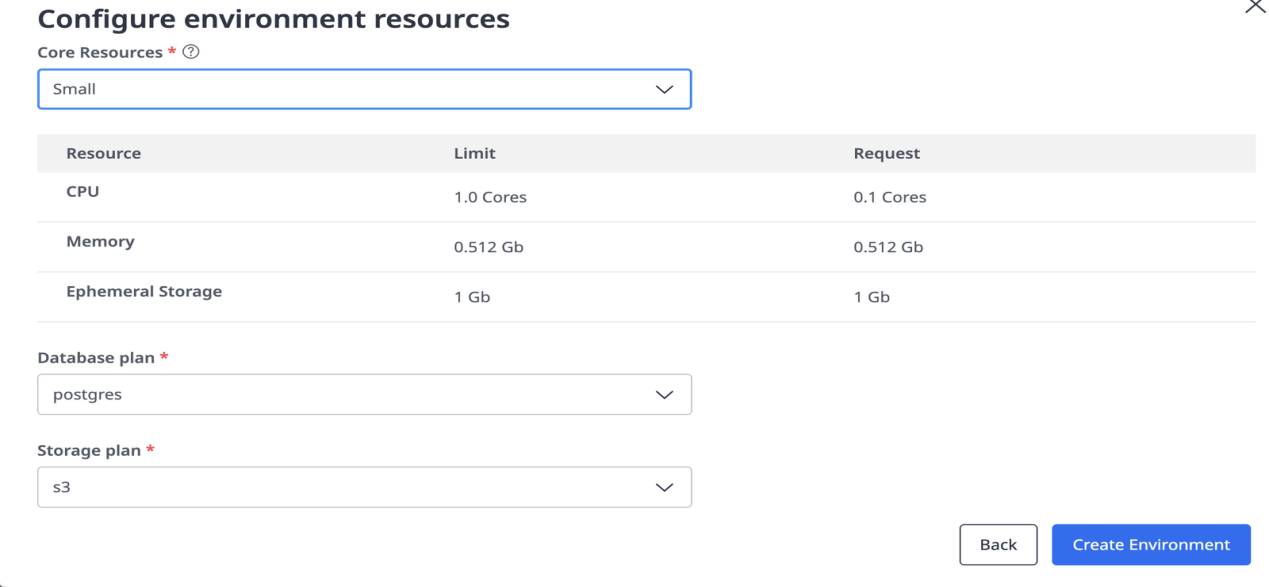
已配置完成 点击 Exit Insaller 退出

## 应用部署到私有环境

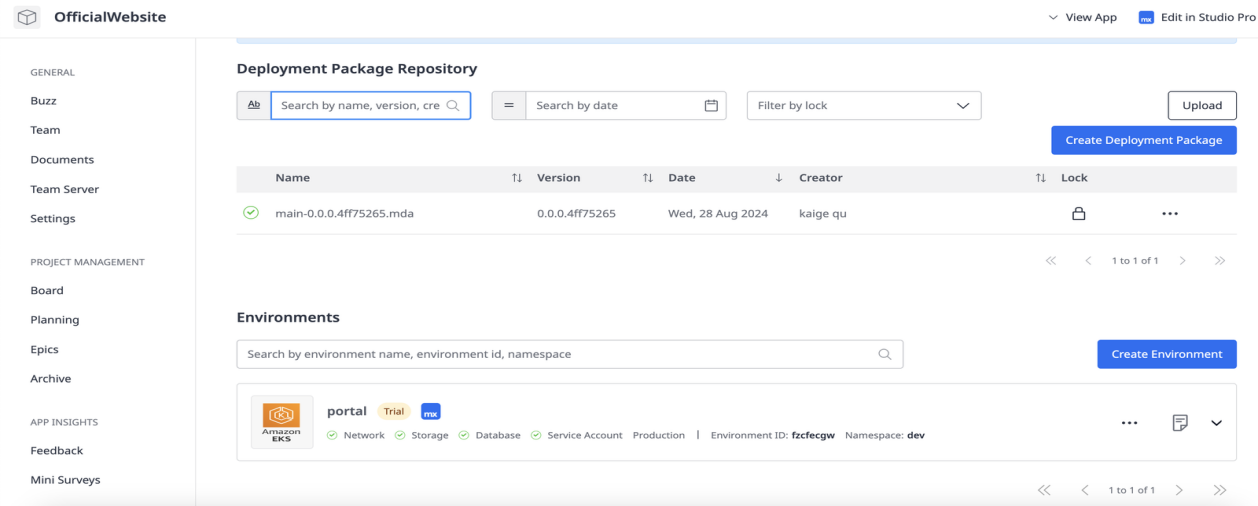


点击Create Environment





创建完成可以看到



通过EKS查看部署的应用



## Mendix数据迁移

官方文档： <https://docs.mendix.com/developerportal/deploy/private-cloud-data-transfer/>

NAMESPACE={namespace}

ENVIRONMENT={environment}

# Create the resources required for the backup operation

kubectl -n $NAMESPACE apply -f /tmp/mendix-backup-restore.yaml

Copy the Linux version of the data migration tool into the Pod

kubectl -n $NAMESPACE cp mxpc-data-migration mendix-backup-restore:/tmp/mxpc-data-migration

# Run the backup process

kubectl -n $NAMESPACE exec -it mendix-backup-restore -- /tmp/mxpc-data-migration backup -e $ENVIRONMENT -f /tmp/backup.tar.gz -n $NAMESPACE

# Copy the backup file from the Pod to a local file

kubectl -n $NAMESPACE cp mendix-backup-restore:/tmp/backup.tar.gz backup.tar.gz

NAMESPACE={new namespace}

ENVIRONMENT={new environment}

# Create the resources required for the restore operation

kubectl -n $NAMESPACE apply -f /tmp/mendix-backup-restore.yaml

# Copy the Linux version of the data migration tool into the Pod

kubectl -n $NAMESPACE cp mxpc-data-migration mendix-backup-restore:/tmp/mxpc-data-migration

# Copy the backup file to be restored into the Pod;# replace files\_and\_database.tar.gz with the path to the backup file

kubectl -n $NAMESPACE cp files\_and\_database.tar.gz mendix-backup-restore:/tmp/restore.tar.gz

# Run the restore process

kubectl -n $NAMESPACE exec -it mendix-backup-restore -- /tmp/mxpc-data-migration restore -e $ENVIRONMENT -f /tmp/restore.tar.gz -n $NAMESPACE

aws s3 cp --recursive s3{bucket}/{prefix} s3://{bucket}/{prefix}

eg: aws s3 cp --recursive s3://365358695168-mx-dev-dev-eip-test/mendix\_qmkqgunr\_file\_582e0c5e\_e367\_4c4f\_b69f\_baa16f53515b/ s3://365358695168-mx-dev-dev-diswgc-mxtestzpj4/mendix\_qmkqgunrttes\_file\_582e0c5e\_e367\_4c4f\_b69f\_baa16f53515b/

## Postgres数据迁移

# 确保可访问到数据库

# 备份

pg\_dump -h 127.0.0.1 -p 5432 -U postgres -c -C -f /data/db.sql dbname

# 恢复

psql -h 127.0.0.1 -p 5432 -U postgres -f /data/dbname.sql

# pg\_dump 数据库备份命令，由postgres提供，默认随数据库安装

# psql 备份恢复命令，由postgres提供，默认随数据库安装

-h 数据库url

-p 端口

-U 用户名

-f 指定路径（导出会自动生成文件）

-c 包含删除表语句（导入时先删除）

-C 包含创建数据库语句（导入时先创建）