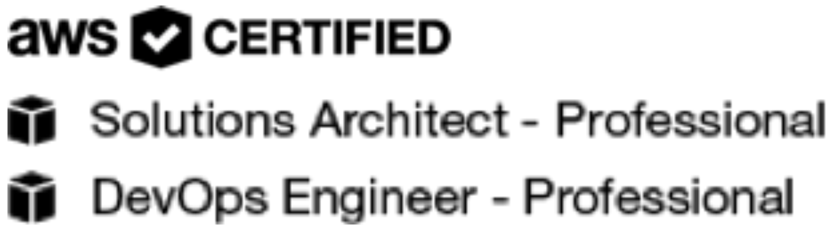
TerraformとWerckerと
AWS Organizationsで始める
ステージング/開発環境構築

- 石澤直人
- 株式会社ヘプタゴン
- インフラエンジニア/クラウドアーキテクト
- golang, python, ruby



aws CERTIFIED

Solutions Architect - Professional

DevOps Engineer - Professional

背景

こんな想いがありました

- 普段インフラエンジニアとしてAWS環境の構築をしている
- いい加減コード化したい
- 構築を自動化したい

=> 当然インフラ側もゴリゴリ検証できる開発環境/ステージング環境必要だよね。Github-Flowのせたいよね。

そのあたりをどう実現するかを

お話します

やりたいこと

- コード化したい
- 構築を自動化したい
- プロダクションとはアカウントレベルで分離された開発/ステージング環境ほしい

どうやるか

- コード化したい

  => **Terraform**

- 構築を自動化したい

  => **Wercker**

- プロダクションとはアカウントレベルで分離された開発/ステージング環境ほしい

  => **AWS Organizations**

Terraform

- AWSでいうCloudformation
- 様々なリソースをコードとして表現できる
- プロバイダーというプラグイン機構を持ち、様々なサービスがプロバイダーとして提供される
- AWSもそのうちの一つ。その他AzureやGCP, Fastly, Github, Herokuなどたくさんある。自作もできる。

Terraform

- Cloudformationを使えばいいじゃん？

  => Cloudformationで扱えるのはAWSリソースだけ。その他のサービスも同等に扱えたほうが管理しやすい。CDNはFastly使いたいとか。

- でも新機能追加されたときの対応速度ってやっぱりCloudformationのほうが早いよね？

  => TerraformからCloudformationを呼べる 🆗

Wercker

- CI as a Service
- CircleCIとかTravisCIの仲間
- 実行環境として独自のDockerコンテナを利用できる
- 複数のsteps(実行すべきコマンドの単位みたいな)から構成されるPipelineがあり、Pipeline単位で環境変数の設定ができる

AWS Organizations

- 複数のAWSアカウントをグループ分けしてポリシーベースで管理できる
- 請求先アカウントもまとめられる
- AWSアカウントをまとめるだけではなく作成も可能
- 現時点で削除はできない

これらを使ってやっていく

アカウントを用意

- AWS Organizationsでサクッと用意する

```
$ aws organizations create-account \
--email '1003ni+aws-private-test2@gmail.com' \
--account-name 'private-test2' \
--role-name 'Naoto_Ishizawa'
{
    "CreateAccountStatus": {
        "RequestedTimestamp": 1498133920.856,
        "State": "IN_PROGRESS",
        "Id": "car-ed0e9d00574411e7bff6500c66d09cc5",
        "AccountName": "private-test2"
    }
}
```

アカウントを用意

```
$ aws organizations list-create-account-status | \
jq '.CreateAccountStatuses[] | \
select(.AccountName == "private-test2")'
{
  "AccountName": "private-test2",
  "State": "SUCCEEDED",
  "RequestedTimestamp": 1498133926.284,
  "CompletedTimestamp": 1498133931.048,
  "Id": "car-ed0e9d00574411e7bff6500c66d09cc5",
  "AccountId": "591019932956"
}
```

アカウントを用意

```
$ cat <<EOL>> ~/.aws/config
[profile private-test2]
role_arn = arn:aws:iam::591019932956:role/Naoto_Ishizawa
source_profile = default
EOL

$ awslogin -profile private-test2
```

- terraform用のIAMユーザーを作成しておく
- tfstateファイル保存用のS3バケット作成しておく

`awslogin` の詳細はこちら

https://blog.youyo.info/post/2017/02/20/create-command-awslogin/

CI環境を整備する

- wercker.yml

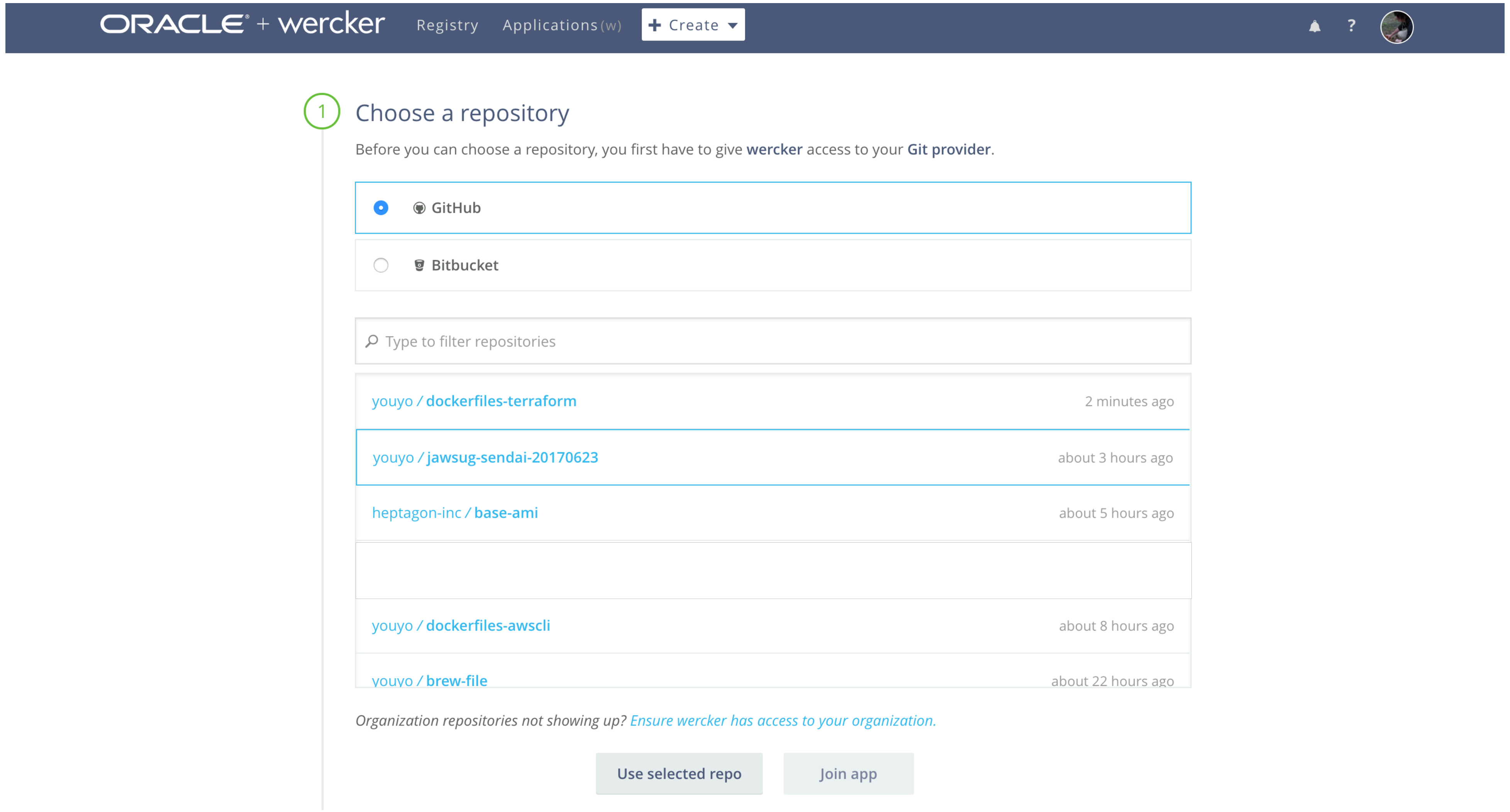
```
start:
  box:
    id: alpine:latest
    cmd: /bin/sh
  steps:
    - script:
      name: Start
      code: |-
        echo "Start ${TF_VAR_env}"
test:
  box:
    id: youyo/terraform:0.9.8
    cmd: /bin/sh
  steps:
    - script:
      name: Test
      code: |-
        echo "Test ${TF_VAR_env}"
        terraform validate
        terraform plan
deploy:
  box:
    id: youyo/terraform:0.9.8
    cmd: /bin/sh
  steps:
    - script:
      name: Deploy
      code: |-
        echo "Deploy ${TF_VAR_env}"
        terraform apply
```

CI環境を整備する

- ブラウザから設定する
    - Applicationの作成
    - Pipeline/Workflowの設定

CI環境を整備する

- Applicationの作成



1 Choose a repository

Before you can choose a repository, you first have to give **wercker** access to your **Git provider**.

● GitHub

○ Bitbucket

Type to filter repositories

youyo / **dockerfiles-terraform** 2 minutes ago

youyo / **jawsug-sendai-20170623** about 3 hours ago

heptagon-inc / **base-ami** about 5 hours ago

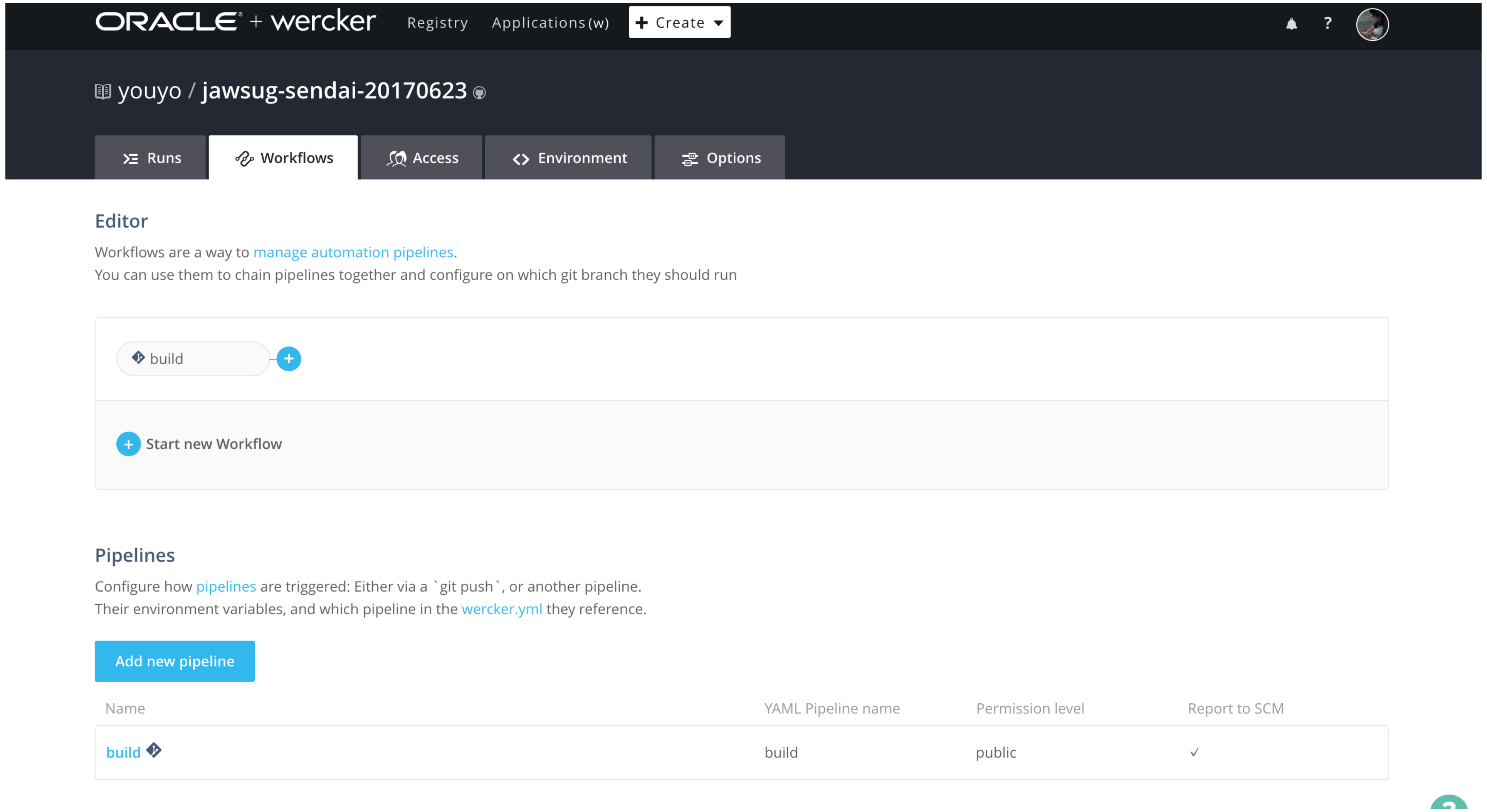
youyo / **dockerfiles-awscli** about 8 hours ago

youyo / **brew-file** about 22 hours ago

Organization repositories not showing up? Ensure wercker has access to your organization.

CI環境を整備する

- Pipeline/Workflowの設定



Editor

Workflows are a way to manage automation pipelines.
You can use them to chain pipelines together and configure on which git branch they should run

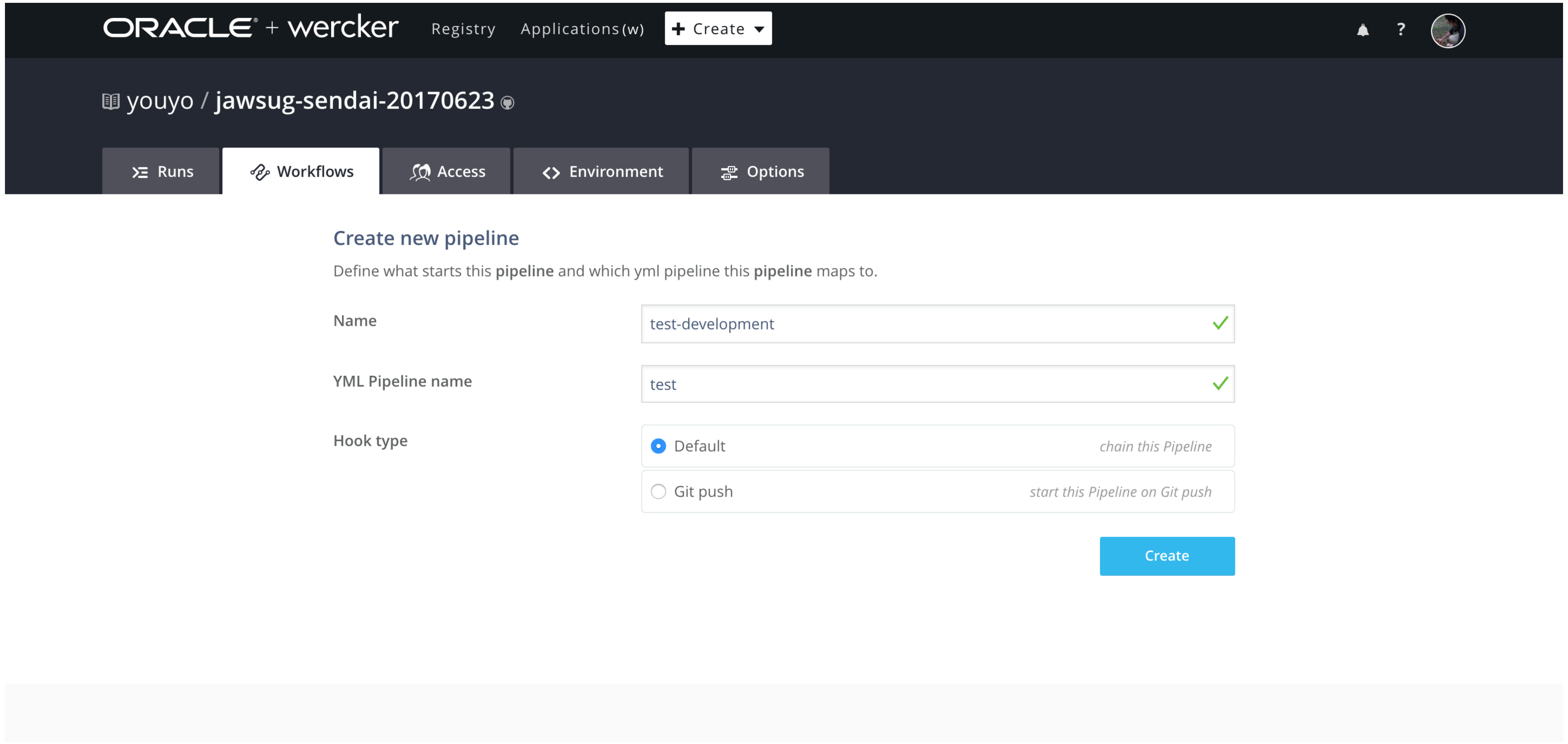
Pipelines

Configure how pipelines are triggered: Either via a `git push`, or another pipeline.
Their environment variables, and which pipeline in the wercker.yml they reference.

Name	YAML Pipeline name	Permission level	Report to SCM
build	build	public	✓

CI環境を整備する

- Pipeline/Workflowの設定



Create new pipeline

Define what starts this **pipeline** and which yml pipeline this **pipeline** maps to.

Name: test-development ✓

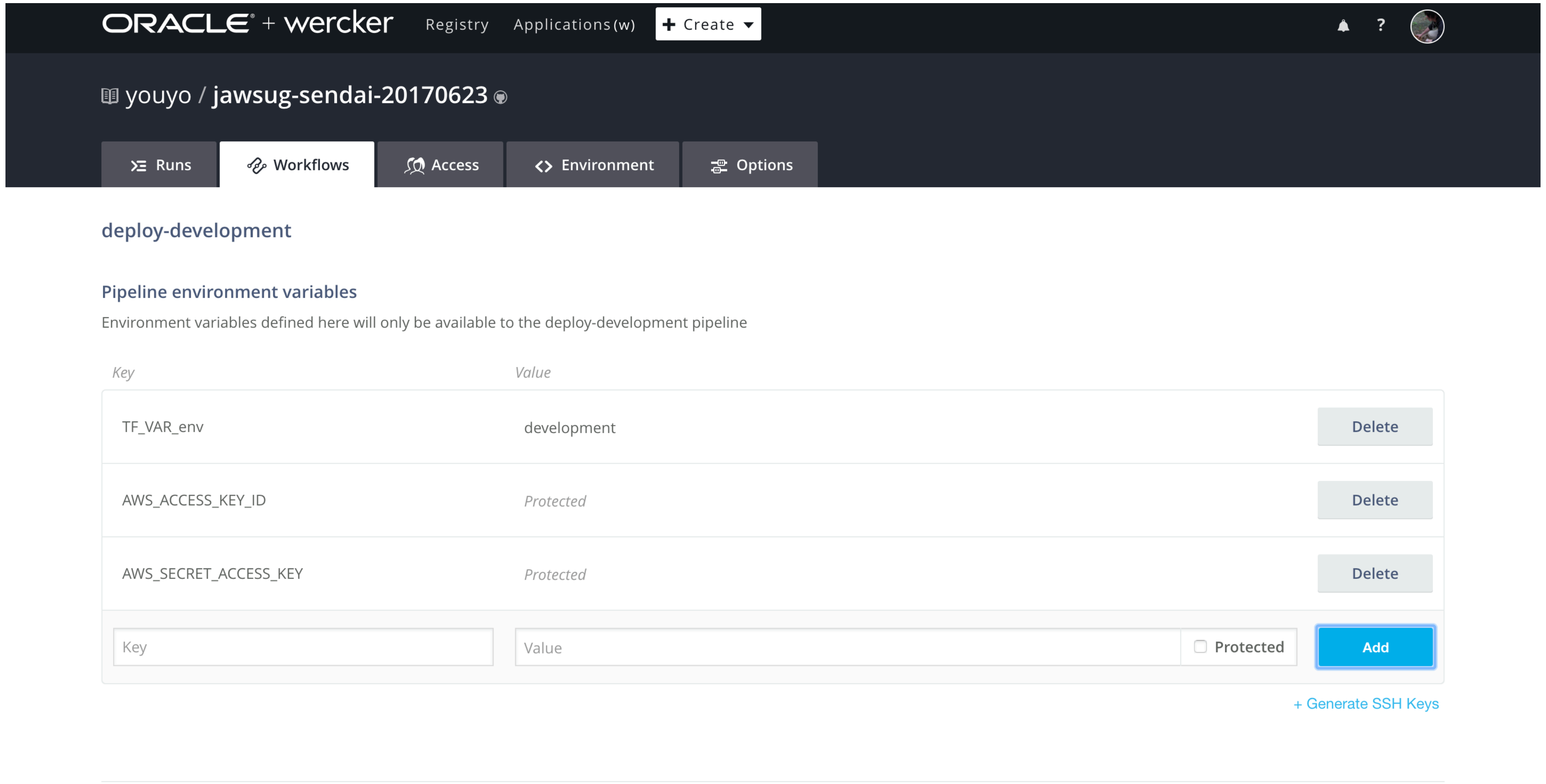
YML Pipeline name: test ✓

Hook type:
● Default *chain this Pipeline*
○ Git push *start this Pipeline on Git push*

Create

CI環境を整備する

- Pipeline/Workflowの設定



deploy-development

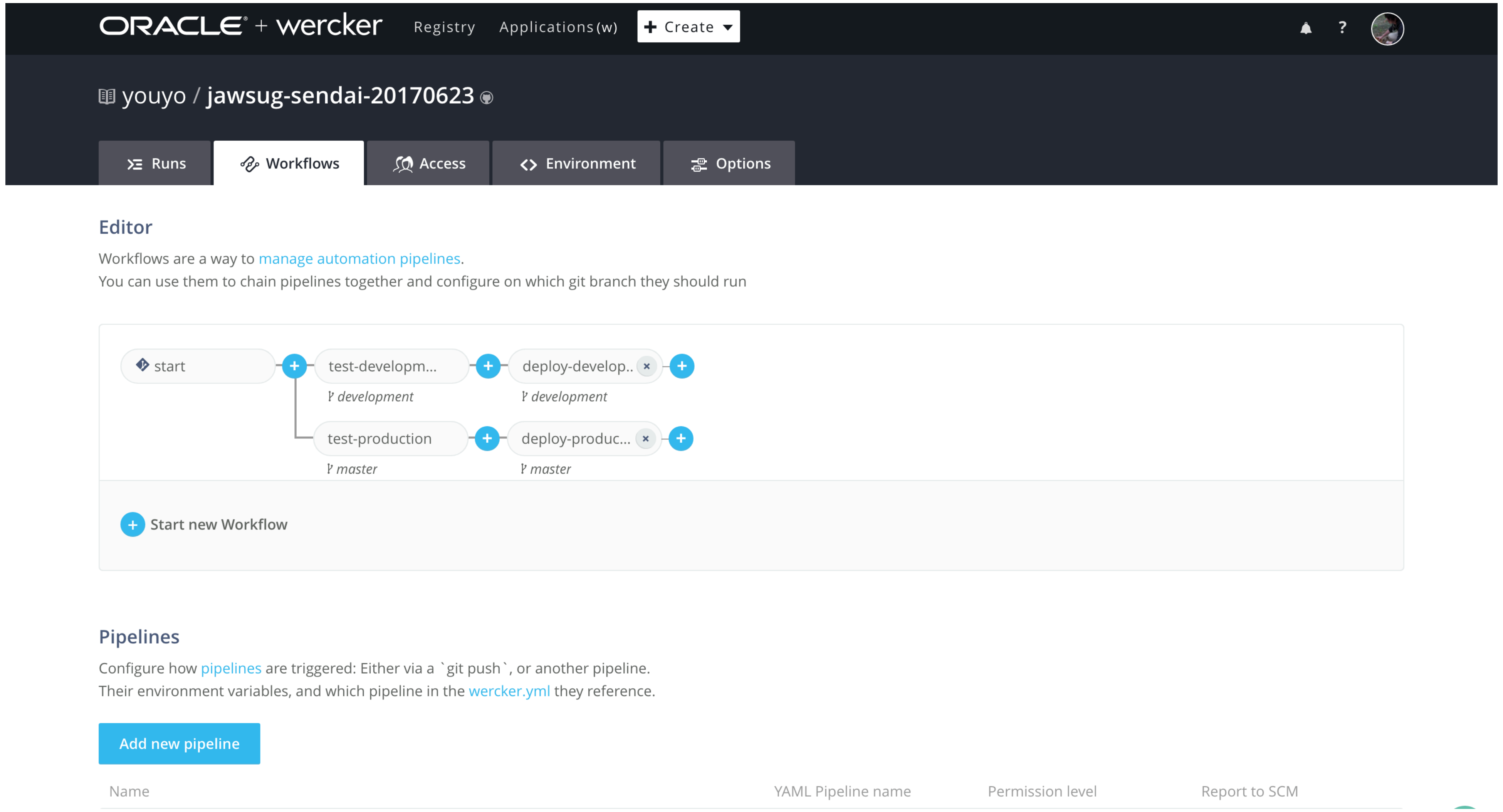
Pipeline environment variables

Environment variables defined here will only be available to the deploy-development pipeline

Key	Value	
TF_VAR_env	development	Delete
AWS_ACCESS_KEY_ID	*Protected*	Delete
AWS_SECRET_ACCESS_KEY	*Protected*	Delete
Key	Value ☐ Protected	Add

CI環境を整備する

- Pipeline/Workflowの設定



Editor

Workflows are a way to manage automation pipelines.
You can use them to chain pipelines together and configure on which git branch they should run

Pipelines

Configure how pipelines are triggered: Either via a `git push`, or another pipeline.
Their environment variables, and which pipeline in the wercker.yml they reference.

Name	YAML Pipeline name	Permission level	Report to SCM

CI環境を整備する

- 各環境ごとにpipelineを用意する
- pipelineごとに環境変数が設定可能

  => commitするbranchによって使用するawsアカウントの切り替えが可能
- `wercker.yml` に変数を記述することもできるがアクセスキーなどの秘匿情報はサイト上から設定することで隠蔽できる

コードを書く

コードを書く

- ログ保存用のS3バケットを作成し
- cloudtrailを設定してみる

コードを書く

```
terraform {
  backend "s3" {
    key    = "terraform.tfstate"
    region = "ap-northeast-1"
  }
}

variable "aws_region" {
  default = "ap-northeast-1"
}

variable "env" {
  default = "development"
}

provider "aws" {
  region = "${var.aws_region}"
}
```

コードを書く

```
resource "aws_s3_bucket" "log" {
  bucket        = "${var.env}-log-jawsug-sendai"
  region        = "${var.aws_region}"
  acl           = "log-delivery-write"
  force_destroy = true
}

resource "aws_s3_bucket_policy" "log" {
  bucket     = "${aws_s3_bucket.log.id}"
  policy     = "${data.aws_iam_policy_document.log.json}"
  depends_on = ["aws_s3_bucket.log"]
}
```

コードを書く

```
data "aws_iam_policy_document" "log" {
  statement {
    sid       = "AWSCloudTrailAclCheck"
    effect    = "Allow"
    actions   = ["s3:GetBucketAcl"]
    resources = ["arn:aws:s3:::${aws_s3_bucket.log.id}"]

    principals {
      type        = "Service"
      identifiers = ["cloudtrail.amazonaws.com"]
    }
  }

  statement {
    sid       = "AWSCloudTrailWrite"
    effect    = "Allow"
    actions   = ["s3:PutObject"]
    resources = ["arn:aws:s3:::${aws_s3_bucket.log.id}/*"]

    principals {
      type        = "Service"
      identifiers = ["cloudtrail.amazonaws.com"]
    }

    condition {
      test     = "StringEquals"
      variable = "s3:x-amz-acl"
      values   = ["bucket-owner-full-control"]
    }
  }
}
```

コードを書く

```
resource "aws_cloudtrail" "cloudtrail" {
  name                          = "cloudtrail"
  enable_logging                = true
  s3_bucket_name                = "${aws_s3_bucket.log.id}"
  include_global_service_events = true
  is_multi_region_trail         = true
  enable_log_file_validation    = true
  depends_on                    = ["aws_s3_bucket_policy.log"]
}
```

コードを書く

- 手元で動かしてみる

コードを書く

- Initialize

```
$ export AWS_ACCESS_KEY_ID=xxx
$ export AWS_SECRET_ACCESS_KEY=yyy
$ export TF_VAR_env=development

$ terraform init -backend=true \
-backend-config="bucket=${TF_VAR_env}-jawsug-sendai-terraform-tfstate"
```

コードを書く

- シンタックスチェックしてみる

```
$ terraform validate
```

何も出力されなければok.

コードを書く

- 実行計画見てみる

```
$ terraform plan
```

コードを書く

```
Refreshing Terraform state in-memory prior to plan...
The refreshed state will be used to calculate this plan, but will not be
persisted to local or remote state storage.

The Terraform execution plan has been generated and is shown below.
Resources are shown in alphabetical order for quick scanning. Green resources
will be created (or destroyed and then created if an existing resource
exists), yellow resources are being changed in-place, and red resources
will be destroyed. Cyan entries are data sources to be read.

Note: You didn't specify an "-out" parameter to save this plan, so when
"apply" is called, Terraform can't guarantee this is what will execute.

+ aws_cloudtrail.cloudtrail
    arn:                           "<computed>"
    enable_log_file_validation:    "true"
    enable_logging:                "true"
    home_region:                   "<computed>"
    include_global_service_events: "true"
    is_multi_region_trail:         "true"
    name:                          "cloudtrail"
    s3_bucket_name:                "${aws_s3_bucket.log.id}"

+ aws_s3_bucket.log
    acceleration_status: "<computed>"
    acl:                 "log-delivery-write"
    arn:                 "<computed>"
    bucket:              "development-log-jawsug-sendai"
    bucket_domain_name:  "<computed>"
    force_destroy:       "true"
    hosted_zone_id:      "<computed>"
    region:              "ap-northeast-1"
    request_payer:       "<computed>"
    versioning.#:        "<computed>"
    website_domain:      "<computed>"
    website_endpoint:    "<computed>"

+ aws_s3_bucket_policy.log
    bucket: "${aws_s3_bucket.log.id}"
    policy: "${data.aws_iam_policy_document.log.json}"

<= data.aws_iam_policy_document.log
    json:                                                    "<computed>"
    statement.#:                                             "2"
    statement.0.actions.#:                                   "1"
    statement.0.actions.845481387:                           "s3:GetBucketAcl"
    statement.0.effect:                                      "Allow"
    statement.0.principals.#:                                "1"
    statement.0.principals.888245936.identifiers.#:          "1"
    statement.0.principals.888245936.identifiers.4136227984: "cloudtrail.amazonaws.com"
    statement.0.principals.888245936.type:                   "Service"
    statement.0.resources.#:                                 "<computed>"
    statement.0.sid:                                         "AWSCloudTrailAclCheck"
    statement.1.actions.#:                                   "1"
    statement.1.actions.315547055:                           "s3:PutObject"
    statement.1.condition.#:                                 "1"
    statement.1.condition.745978174.test:                    "StringEquals"
    statement.1.condition.745978174.values.#:                "1"
    statement.1.condition.745978174.values.544818230:        "bucket-owner-full-control"
    statement.1.condition.745978174.variable:                "s3:x-amz-acl"
    statement.1.effect:                                      "Allow"
    statement.1.principals.#:                                "1"
    statement.1.principals.888245936.identifiers.#:          "1"
    statement.1.principals.888245936.identifiers.4136227984: "cloudtrail.amazonaws.com"
    statement.1.principals.888245936.type:                   "Service"
    statement.1.resources.#:                                 "<computed>"
    statement.1.sid:                                         "AWSCloudTrailWrite"


Plan: 3 to add, 0 to change, 0 to destroy.
```

コードを書く

- 3つのリソースが追加される

```
Plan: 3 to add, 0 to change, 0 to destroy.
```

コードを書く

- applyする

```
$ terraform apply

aws_s3_bucket.log: Creating...
  acceleration_status: "" => "<computed>"
  acl:                 "" => "log-delivery-write"
  arn:                 "" => "<computed>"
  bucket:              "" => "development-log-jawsug-sendai"
  bucket_domain_name:  "" => "<computed>"
  force_destroy:       "" => "true"
  hosted_zone_id:      "" => "<computed>"
  region:              "" => "ap-northeast-1"
  request_payer:       "" => "<computed>"
  versioning.#:        "" => "<computed>"
  website_domain:      "" => "<computed>"
  website_endpoint:    "" => "<computed>"
aws_s3_bucket.log: Creation complete (ID: development-log-jawsug-sendai)
data.aws_iam_policy_document.log: Refreshing state...
aws_s3_bucket_policy.log: Creating...
  bucket: "" => "development-log-jawsug-sendai"
  policy: "" => "省略"
aws_s3_bucket_policy.log: Creation complete (ID: development-log-jawsug-sendai)
aws_cloudtrail.cloudtrail: Creating...
  arn:                           "" => "<computed>"
  enable_log_file_validation:    "" => "true"
  enable_logging:                "" => "true"
  home_region:                   "" => "<computed>"
  include_global_service_events: "" => "true"
  is_multi_region_trail:         "" => "true"
  name:                          "" => "cloudtrail"
  s3_bucket_name:                "" => "development-log-jawsug-sendai"
aws_cloudtrail.cloudtrail: Creation complete (ID: cloudtrail)

Apply complete! Resources: 3 added, 0 changed, 0 destroyed.

The state of your infrastructure has been saved to the path
below. This state is required to modify and destroy your
infrastructure, so keep it safe. To inspect the complete state
use the `terraform show` command.

State path:
```

コードを書く

- showしてみる

```
$ terraform show

aws_cloudtrail.cloudtrail:
  id = cloudtrail
  arn = arn:aws:cloudtrail:ap-northeast-1:591019932956:trail/cloudtrail
  cloud_watch_logs_group_arn =
  cloud_watch_logs_role_arn =
  enable_log_file_validation = true
  enable_logging = true
  home_region = ap-northeast-1
  include_global_service_events = true
  is_multi_region_trail = true
  kms_key_id =
  name = cloudtrail
  s3_bucket_name = development-log-jawsug-sendai
  s3_key_prefix =
  sns_topic_name =
  tags.% = 0
aws_s3_bucket.log:
  id = development-log-jawsug-sendai
  acceleration_status =
  acl = log-delivery-write
  arn = arn:aws:s3:::development-log-jawsug-sendai
  bucket = development-log-jawsug-sendai
  bucket_domain_name = development-log-jawsug-sendai.s3.amazonaws.com
  force_destroy = true
  hosted_zone_id = Z2M4EHUR26P7ZW
  logging.# = 0
  region = ap-northeast-1
  request_payer = BucketOwner
  tags.% = 0
  versioning.# = 1
  versioning.0.enabled = false
  versioning.0.mfa_delete = false
  website.# = 0
aws_s3_bucket_policy.log:
  id = development-log-jawsug-sendai
  bucket = development-log-jawsug-sendai
  policy = {
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AWSCloudTrailAclCheck",
      "Effect": "Allow",
      "Action": "s3:GetBucketAcl",
      "Resource": "arn:aws:s3:::development-log-jawsug-sendai",
      "Principal": {
        "Service": "cloudtrail.amazonaws.com"
      }
    },
    {
      "Sid": "AWSCloudTrailWrite",
      "Effect": "Allow",
      "Action": "s3:PutObject",
      "Resource": "arn:aws:s3:::development-log-jawsug-sendai/*",
      "Principal": {
        "Service": "cloudtrail.amazonaws.com"
      },
      "Condition": {
        "StringEquals": {
          "s3:x-amz-acl": "bucket-owner-full-control"
        }
      }
    }
  ]
}
data.aws_iam_policy_document.log:
  id = 3591822430
  json = {
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AWSCloudTrailAclCheck",
      "Effect": "Allow",
      "Action": "s3:GetBucketAcl",
      "Resource": "arn:aws:s3:::development-log-jawsug-sendai",
      "Principal": {
        "Service": "cloudtrail.amazonaws.com"
      }
    },
    {
      "Sid": "AWSCloudTrailWrite",
      "Effect": "Allow",
      "Action": "s3:PutObject",
      "Resource": "arn:aws:s3:::development-log-jawsug-sendai/*",
      "Principal": {
        "Service": "cloudtrail.amazonaws.com"
      },
      "Condition": {
        "StringEquals": {
          "s3:x-amz-acl": "bucket-owner-full-control"
        }
      }
    }
  ]
}
  statement.# = 2
  statement.0.actions.# = 1
  statement.0.actions.845481387 = s3:GetBucketAcl
  statement.0.condition.# = 0
  statement.0.effect = Allow
  statement.0.not_actions.# = 0
  statement.0.not_principals.# = 0
  statement.0.not_resources.# = 0
  statement.0.principals.# = 1
  statement.0.principals.888245936.identifiers.# = 1
  statement.0.principals.888245936.identifiers.4136227984 = cloudtrail.amazonaws.com
  statement.0.principals.888245936.type = Service
  statement.0.resources.# = 1
  statement.0.resources.2709316599 = arn:aws:s3:::development-log-jawsug-sendai
  statement.0.sid = AWSCloudTrailAclCheck
  statement.1.actions.# = 1
  statement.1.actions.315547055 = s3:PutObject
  statement.1.condition.# = 1
  statement.1.condition.745978174.test = StringEquals
  statement.1.condition.745978174.values.# = 1
  statement.1.condition.745978174.values.544818230 = bucket-owner-full-control
  statement.1.condition.745978174.variable = s3:x-amz-acl
  statement.1.effect = Allow
  statement.1.not_actions.# = 0
  statement.1.not_principals.# = 0
  statement.1.not_resources.# = 0
  statement.1.principals.# = 1
  statement.1.principals.888245936.identifiers.# = 1
  statement.1.principals.888245936.identifiers.4136227984 = cloudtrail.amazonaws.com
  statement.1.principals.888245936.type = Service
  statement.1.resources.# = 1
  statement.1.resources.3865598363 = arn:aws:s3:::development-log-jawsug-sendai/*
  statement.1.sid = AWSCloudTrailWrite
```

あとはこれらがCIサービス(wercker)
上で動けばいい

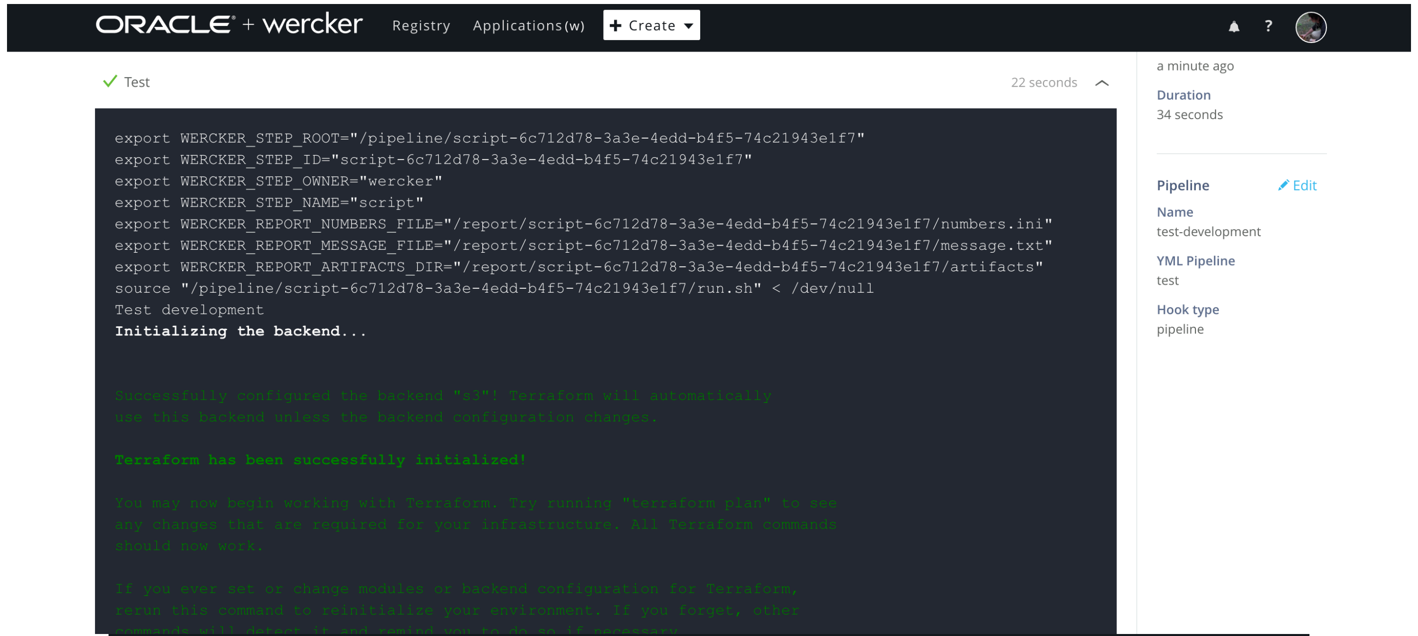
祈りながら `git push` 🙏

祈りながら git push 🙏

- ちょっと変更加えてpush

```
$ git diff
diff --git a/s3.tf b/s3.tf
index 2422f44..136743b 100644
--- a/s3.tf
+++ b/s3.tf
@@ -42,3 +42,10 @@ resource "aws_s3_bucket_policy" "log" {
   policy     = "${data.aws_iam_policy_document.log.json}"
   depends_on = ["aws_s3_bucket.log"]
 }
+
+resource "aws_s3_bucket" "test" {
+  bucket        = "${var.env}-test-jawsug-sendai"
+  region        = "${var.aws_region}"
+  acl           = "private"
+  force_destroy = true
+}
```

祈りながら git push 🙏



```
export WERCKER_STEP_ROOT="/pipeline/script-6c712d78-3a3e-4edd-b4f5-74c21943e1f7"
export WERCKER_STEP_ID="script-6c712d78-3a3e-4edd-b4f5-74c21943e1f7"
export WERCKER_STEP_OWNER="wercker"
export WERCKER_STEP_NAME="script"
export WERCKER_REPORT_NUMBERS_FILE="/report/script-6c712d78-3a3e-4edd-b4f5-74c21943e1f7/numbers.ini"
export WERCKER_REPORT_MESSAGE_FILE="/report/script-6c712d78-3a3e-4edd-b4f5-74c21943e1f7/message.txt"
export WERCKER_REPORT_ARTIFACTS_DIR="/report/script-6c712d78-3a3e-4edd-b4f5-74c21943e1f7/artifacts"
source "/pipeline/script-6c712d78-3a3e-4edd-b4f5-74c21943e1f7/run.sh" < /dev/null
Test development
Initializing the backend...


Successfully configured the backend "s3"! Terraform will automatically
use this backend unless the backend configuration changes.

Terraform has been successfully initialized!

You may now begin working with Terraform. Try running "terraform plan" to see
any changes that are required for your infrastructure. All Terraform commands
should now work.

If you ever set or change modules or backend configuration for Terraform,
rerun this command to reinitialize your environment. If you forget, other
commands will detect it and remind you to do so if necessary.
```

a minute ago

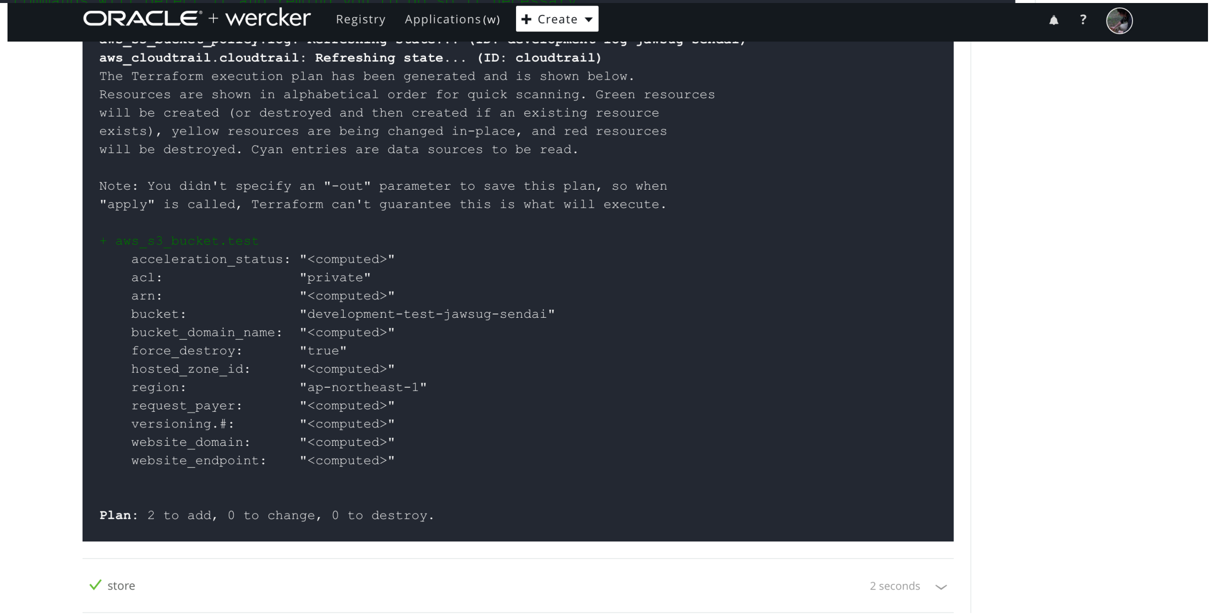
Duration
34 seconds

Pipeline ✎ Edit

Name
test-development

YML Pipeline
test

Hook type
pipeline



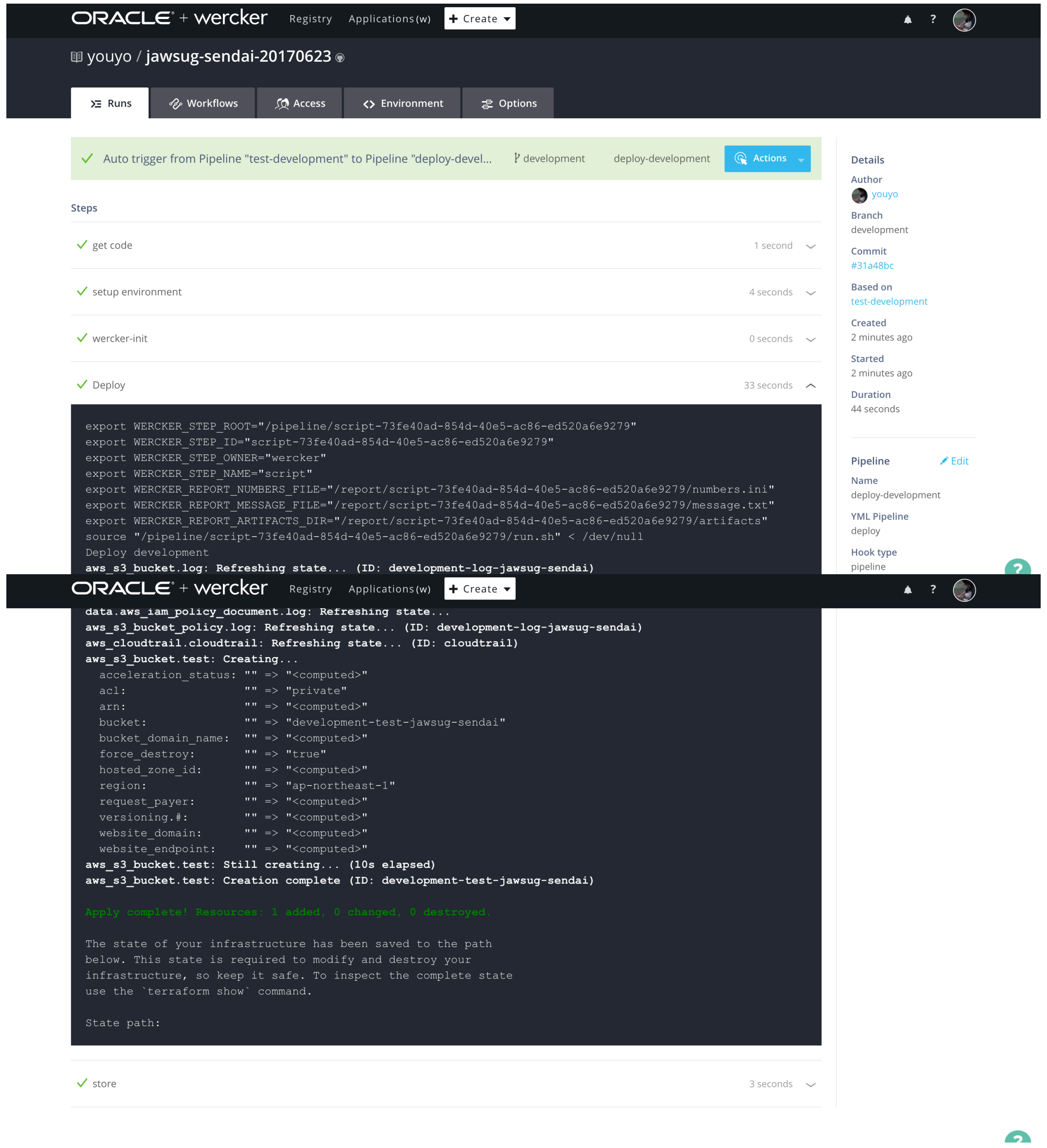
```
aws_s3_bucket_policy.log: Refreshing state... (ID: development-log-jawsug-sendai)
aws_cloudtrail.cloudtrail: Refreshing state... (ID: cloudtrail)
The Terraform execution plan has been generated and is shown below.
Resources are shown in alphabetical order for quick scanning. Green resources
will be created (or destroyed and then created if an existing resource
exists), yellow resources are being changed in-place, and red resources
will be destroyed. Cyan entries are data sources to be read.

Note: You didn't specify an "-out" parameter to save this plan, so when
"apply" is called, Terraform can't guarantee this is what will execute.

+ aws_s3_bucket.test
    acceleration_status: "<computed>"
    acl:                 "private"
    arn:                 "<computed>"
    bucket:              "development-test-jawsug-sendai"
    bucket_domain_name:  "<computed>"
    force_destroy:       "true"
    hosted_zone_id:      "<computed>"
    region:              "ap-northeast-1"
    request_payer:       "<computed>"
    versioning.#:        "<computed>"
    website_domain:      "<computed>"
    website_endpoint:    "<computed>"


Plan: 2 to add, 0 to change, 0 to destroy.
```

祈りながら git push 🙏



✓ Auto trigger from Pipeline "test-development" to Pipeline "deploy-devel... development deploy-development

Steps

✓ get code 1 second

✓ setup environment 4 seconds

✓ wercker-init 0 seconds

✓ Deploy 33 seconds

```
export WERCKER_STEP_ROOT="/pipeline/script-73fe40ad-854d-40e5-ac86-ed520a6e9279"
export WERCKER_STEP_ID="script-73fe40ad-854d-40e5-ac86-ed520a6e9279"
export WERCKER_STEP_OWNER="wercker"
export WERCKER_STEP_NAME="script"
export WERCKER_REPORT_NUMBERS_FILE="/report/script-73fe40ad-854d-40e5-ac86-ed520a6e9279/numbers.ini"
export WERCKER_REPORT_MESSAGE_FILE="/report/script-73fe40ad-854d-40e5-ac86-ed520a6e9279/message.txt"
export WERCKER_REPORT_ARTIFACTS_DIR="/report/script-73fe40ad-854d-40e5-ac86-ed520a6e9279/artifacts"
source "/pipeline/script-73fe40ad-854d-40e5-ac86-ed520a6e9279/run.sh" < /dev/null
Deploy development
aws_s3_bucket.log: Refreshing state... (ID: development-log-jawsug-sendai)
data.aws_iam_policy_document.log: Refreshing state...
aws_s3_bucket_policy.log: Refreshing state... (ID: development-log-jawsug-sendai)
aws_cloudtrail.cloudtrail: Refreshing state... (ID: cloudtrail)
aws_s3_bucket.test: Creating...
  acceleration_status: "" => "<computed>"
  acl:                 "" => "private"
  arn:                 "" => "<computed>"
  bucket:              "" => "development-test-jawsug-sendai"
  bucket_domain_name:  "" => "<computed>"
  force_destroy:       "" => "true"
  hosted_zone_id:      "" => "<computed>"
  region:              "" => "ap-northeast-1"
  request_payer:       "" => "<computed>"
  versioning.#:        "" => "<computed>"
  website_domain:      "" => "<computed>"
  website_endpoint:    "" => "<computed>"
aws_s3_bucket.test: Still creating... (10s elapsed)
aws_s3_bucket.test: Creation complete (ID: development-test-jawsug-sendai)

Apply complete! Resources: 1 added, 0 changed, 0 destroyed.

The state of your infrastructure has been saved to the path
below. This state is required to modify and destroy your
infrastructure, so keep it safe. To inspect the complete state
use the `terraform show` command.

State path:
```

✓ store 3 seconds

Details

Author
youyo

Branch
development

Commit
#31a48bc

Based on
test-development

Created
2 minutes ago

Started
2 minutes ago

Duration
44 seconds

Pipeline Edit

Name
deploy-development

YML Pipeline
deploy

Hook type
pipeline

無事自動化された開発環境を手に入れました

Github-Flow的なことも

できるようになったし

どんどん開発していくぞ！！！

まとめ

- AWS Organizations/Terraform/Werckerなどを組み合わせてコード化/自動化できた
- ここはあくまでいい感じのスタート地点
- 生産性高く開発していこう
- 本日の資料はこちらに

  https://github.com/youyo/jawsug-sendai-20170623