

Terraform

Hva, hvorfor og hvordan

Workshop @ Amedia

2024-05-28

Agenda

- Hvordan lager vi ressurser i skyen?
- Hva er Terraform?
- Hvorfor er det lurt å bruke det?
- Hvordan fungerer det egentlig?
- Vanlige operasjoner
- Spesialoperasjoner



HashiCorp

Terraform

Hvordan lager vi ressurser i skyen?

ClickOps™

Den raskeste veien til mål

[video](#)

Hvordan lager vi ressurser i skyen?

CLI

Presist

```
gcloud --project='amedia-adp-test' pubsub \
  topics create 'my-topic' \
  --message-retention-duration=1d
```

```
gcloud --project='amedia-adp-test' pubsub \
  subscriptions create 'my-subscription' \
  --topic='my-topic'
```

Hvordan lager vi ressurser i skyen?

CLI

Persist

```
gcloud --project='amedia-adp-test' pubsub \
  topics create 'my-topic' \
  --message-retention-duration=1d
```

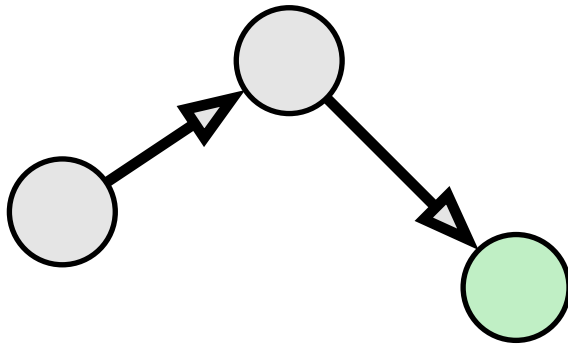
```
gcloud --project='amedia-adp-test' pubsub \
  subscriptions create 'my-subscription' \
  --topic='my-topic'
```

[create-infrastructure.sh](#)

Fellestrekk

ClickOps og gcloud

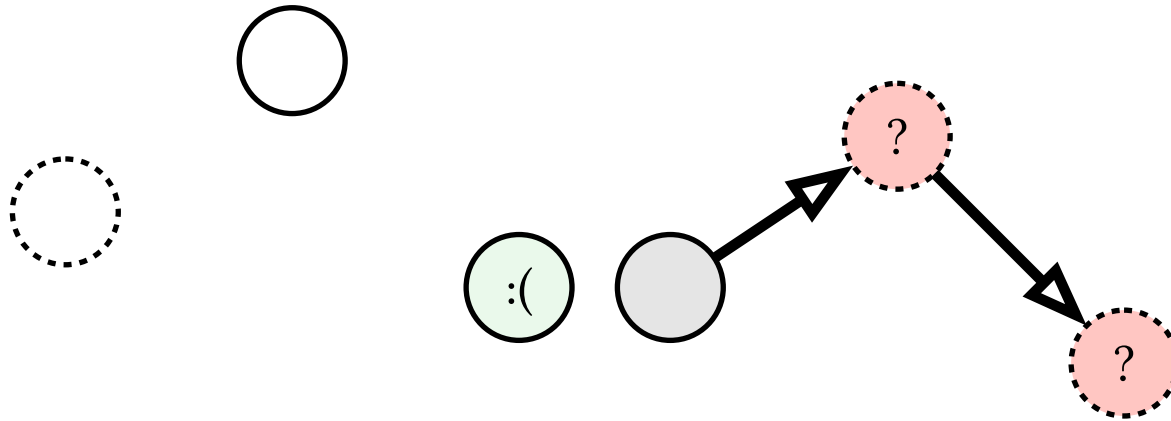
- Sekvens av steg som forhåpentligvis tar deg til mål
- Beskriver handlinger, ikke tilstand
- Er ikke «idempotent»



Fellestrekke

ClickOps og gcloud

- Sekvens av steg som forhåpentligvis tar deg til mål
- Beskriver handlinger, ikke tilstand
- Er ikke «idempotent»



Hva er Terraform?

infrastruktur som kode
(infrastructure as code, IaC)

Ønsket tilstand

Jeg vil ha:

- Et pubsub topic som heter my - fancy - topic
- En subscription som heter my - fancy - topic - subscription
 - ack-deadline på 20 sekunder
 - pusher til et endepunkt

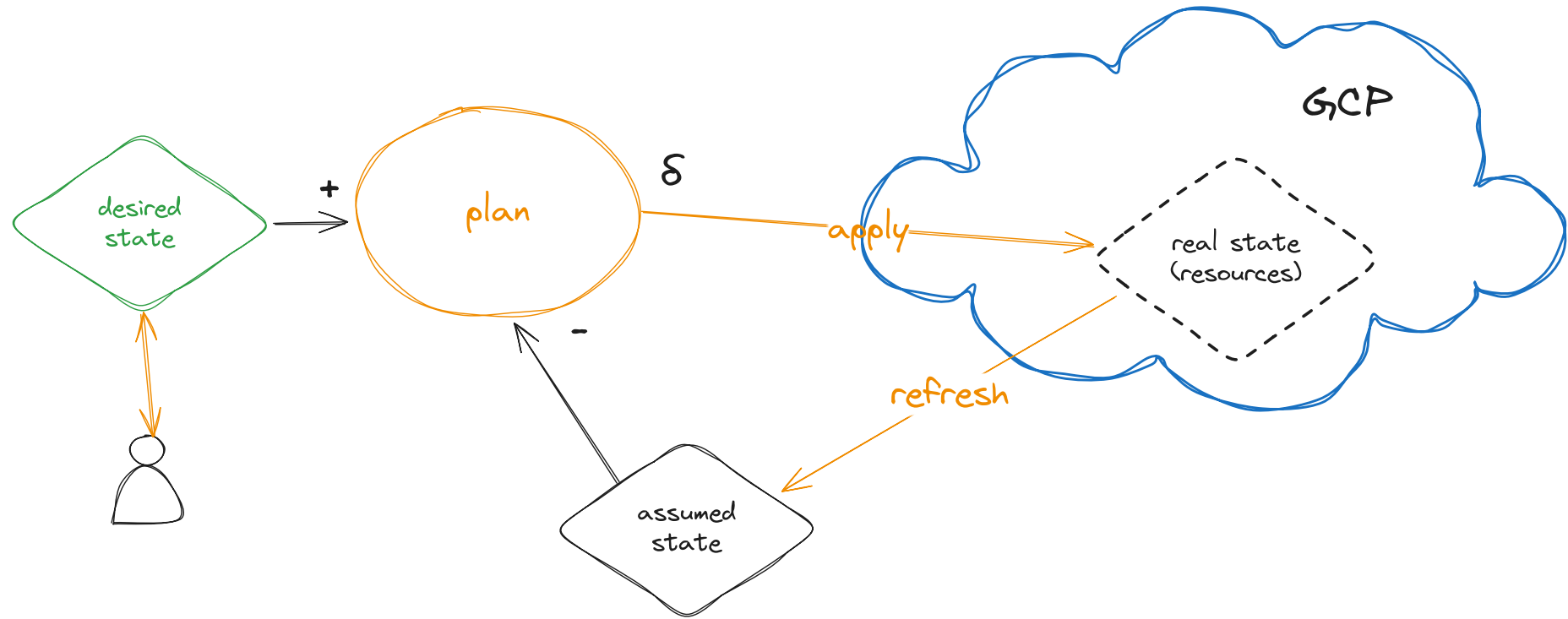
Ønsket tilstand

Jeg vil ha:

- Et pubsub topic som heter my-fancy-topic
- En subscription som heter my-fancy-topic-subscription
 - ack-deadline på 20 sekunder
 - pusher til et endepunkt

```
resource "google_pubsub_topic" "the-topic" {  
  name = "my-fancy-topic"  
}  
  
resource "google_pubsub_subscription" "the-subscription" {  
  name = "my-fancy-topic-subscription"  
  topic = google_pubsub_topic.the-topic.name  
  ack_deadline_seconds = 20  
  push_config {  
    push_endpoint = "https://my-endpoint.example.com/notify"  
  }  
}
```

Hva er Terraform?



Hvorfor bruke Terraform?

Hvorfor bruke Terraform?

- koden *er* infrastrukturen \implies «dokumentasjonen» vedlikeholdes automatisk
- historikk ved hjelp av git
- tjenestene blir mer reproduserbare

Hvorfor bruke Terraform?

- koden *er* infrastrukturen \implies «dokumentasjonen» vedlikeholdes automatisk
- historikk ved hjelp av git
- tjenestene blir mer reproduserbare

En tjeneste består av både kode og infrastruktur

Hvordan fungerer Terraform?

Eksempel

```
resource "google_pubsub_topic" "the-topic" {  
  name = "my-fancy-topic"  
}  
  
resource "google_pubsub_subscription" "the-subscription" {  
  name = "my-fancy-topic-subscription"  
  topic = google_pubsub_topic.the-topic.name  
  ack_deadline_seconds = 20  
  push_config {  
    push_endpoint = "https://my-endpoint.example.com/notify"  
  }  
  description = "Subscribes to id ${google_pubsub_topic.the-topic.id}"  
}
```

Eksempel

```
resource "google_pubsub_topic" "the-topic" {  
  name = "my-fancy-topic"  
}
```

ressurstype, navn (scopet), argument

Eksempel

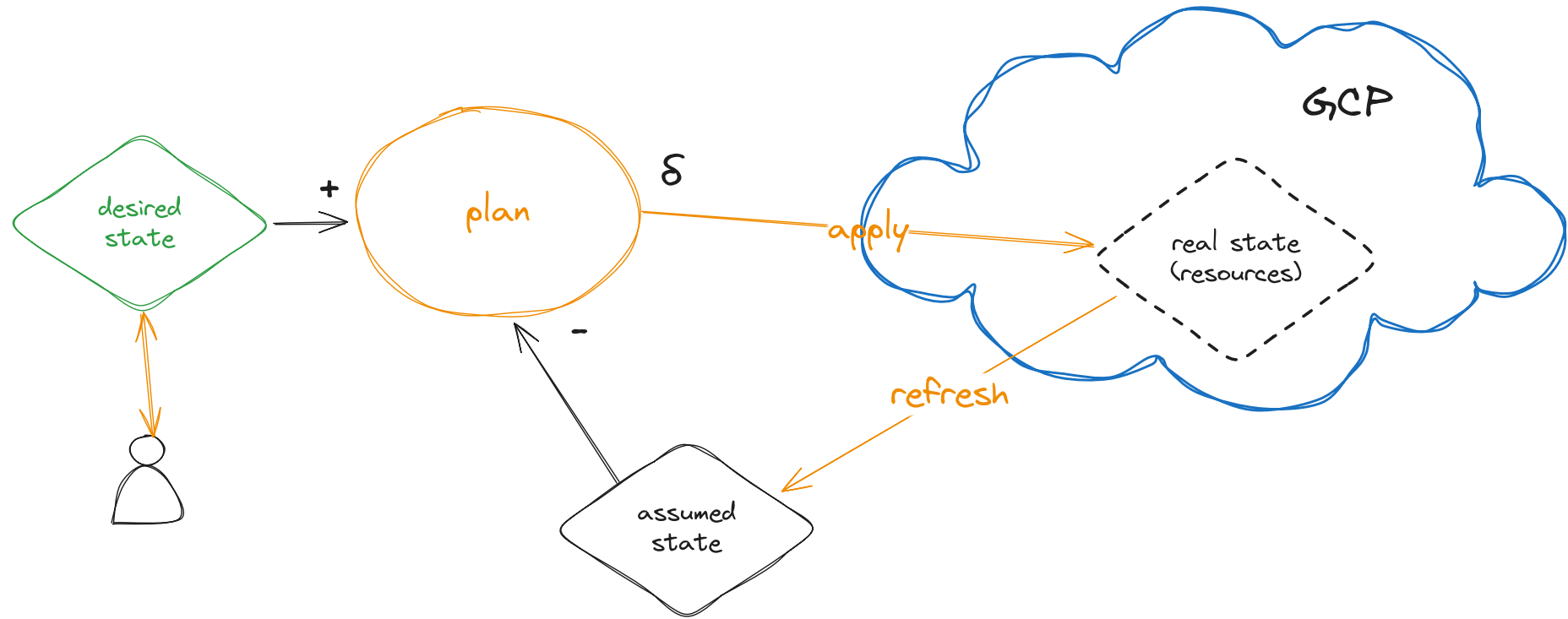
```
resource "google_pubsub_topic" "the-topic" {  
  name = "my-fancy-topic"  
}  
  
resource "google_pubsub_subscription" "the-subscription" {  
  
  topic = google_pubsub_topic.the-topic.name  
  
  description = "Subscribes to id ${google_pubsub_topic.the-topic.id}"  
}
```

referanse til argument og attributt

Eksempel

```
resource "google_pubsub_topic" "the-topic" {  
  name = "my-fancy-topic"  
}  
  
resource "google_pubsub_subscription" "the-subscription" {  
  name = "my-fancy-topic-subscription"  
  topic = google_pubsub_topic.the-topic.name  
  ack_deadline_seconds = 20  
  push_config {  
    push_endpoint = "https://my-endpoint.example.com/notify"  
  }  
  description = "Subscribes to id ${google_pubsub_topic.the-topic.id}"  
}
```

Hvordan fungerer Terraform?



I grove trekk

- utvikleren endrer på en **konfigurasjon** **konfigurasjon** filer som slutter med `.tf` (f.eks. legger til ressurser i `main.tf`)

I grove trekk

- utvikleren endrer på en **konfigurasjon** (f.eks. legger til ressurser i `main.tf`)
- terraform sammenlikner konfigurasjonen med en **tilstand** og lager en **plan**

konfigurasjon filer som slutter med `.tf`
tilstand vanligvis `default.tfstate` – lagret lokalt eller i en bønne (en **backend**) – beskriver hvilke ressurser terraform tracker og tilstanden til disse ressursene – holdes i synk ved hver terraform `plan/refresh`
plan en sekvens av handlinger som utgjør en diff, og fører til at tilstanden er slik konfigurasjonen tilsier

I grove trekk

- utvikleren endrer på en **konfigurasjon** (f.eks. legger til ressurser i `main.tf`)
- terraform sammenlikner konfigurasjonen med en **tilstand** og lager en **plan**
- terraform får tilgang til skyen ved hjelp av en **provider**

konfigurasjon filer som slutter med `.tf`
tilstand vanligvis `default.tfstate` – lagret lokalt eller i en bønne (en **backend**) – beskriver hvilke ressurser terraform tracker og tilstanden til disse ressursene – holdes i synk ved hver terraform `plan/refresh`

plan en sekvens av handlinger som utgjør en diff, og fører til at tilstanden er slik konfigurasjonen tilsier

provider en plugin som beskriver hvilke ressurser som er tilgjengelige, og hvordan de konfigureres

I grove trekk

- utvikleren endrer på en **konfigurasjon** (f.eks. legger til ressurser i `main.tf`)
- terraform sammenlikner konfigurasjonen med en **tilstand** og lager en **plan**
- terraform får tilgang til skyen ved hjelp av en **provider**
- ved hjelp av providerens API-er gjør terraform endringer i ressurser

konfigurasjon filer som slutter med `.tf`

tilstand vanligvis `default.tfstate` – lagret lokalt eller i en bønne (en **backend**) – beskriver hvilke ressurser terraform tracker og tilstanden til disse ressursene – holdes i synk ved hver terraform `plan/refresh`

plan en sekvens av handlinger som utgjør en diff, og fører til at tilstanden er slik konfigurasjonen tilsier

provider en plugin som beskriver hvilke ressurser som er tilgjengelige, og hvordan de konfigureres

I grove trekk

- utvikleren endrer på en **konfigurasjon** (f.eks. legger til ressurser i `main.tf`)
- terraform sammenlikner konfigurasjonen med en **tilstand** og lager en **plan**
- terraform får tilgang til skyen ved hjelp av en **provider**
- ved hjelp av providerens API-er gjør terraform endringer i ressurser
- tilstanden er nå oppdatert slik at den stemmer med konfigurasjonen

konfigurasjon filer som slutter med `.tf`

tilstand vanligvis `default.tfstate` – lagret lokalt eller i en bønne (en **backend**) – beskriver hvilke ressurser terraform tracker og tilstanden til disse ressursene – holdes i synk ved hver terraform `plan/refresh`

plan en sekvens av handlinger som utgjør en diff, og fører til at tilstanden er slik konfigurasjonen tilsier

provider en plugin som beskriver hvilke ressurser som er tilgjengelige, og hvordan de konfigureres