**Presentatie:**

**Slide 1**: voorafgaand aan de presentatie vraag ik iedereen om de repository te clonen. Dat kan zowel via git clone (zie commando op de slide) als via het downloaden van de zip file (ga naar de URL, klik op “Code” en dan “Download Zip file”).

**Slide 5:**

* Resources is vet gedrukt omdat je verplicht bent om die aan te maken

**Slide 7:** demo:

* Ga naar https://eu-west-1.console.aws.amazon.com/console
* Zoek in de bovenste balk CloudFormation
* Laat in de linkerbalk “Stacks” zien
* Create stack kan via de oranje button “Create Stack” als er geen CloudFormation templates zijn, maar deze button is er niet meer zodra er minimaal 1 CloudFormation stack is. De bovenste button “Create Stack” werkt altijd, kies dan de eerste optie “With new resources (standard)”
* Bij “Prepare template” is de eerste optie de juiste, onder “Specify template” kies je voor “Upload a template file”. Daarna het bestand aangeven met je JSON of YAML code. Ik kies voor parameters.yaml uit de github repository.
* Geef een stack name (bijv: parameters-frederique). Pas de BucketNameParameter aan naar de datum van vandaag (20230101 zou al gebruikt kunnen zijn door iemand anders, de kans is klein dat iemand op dezelfde datum dezelfde naam gebruikt)
* De opties in Step 3 zijn meestal goed, dus scroll naar onder en klik op “Next”

(meestal goed: een enkele keer kan het handig zijn om resources niet terug te rollen bij een fout om debugging mogelijk te maken)

* Step 4: scroll naar onder en klik “Submit”
* Wacht tot e.e.a. is uitgerold, leg daarna de tabs “Stack Info”, “Events” (met Logical ID), “Resources” (met zowel Logical ID als Physical ID, klik op de S3 bucket en zie dat je dan in S3 terecht komt), Outputs (niet ingevuld omdat we geen outputs in de template file hebben staan), Parameters, Template. Change sets vallen buiten de scope van deze workshop.

**Slide 8:**

* “Default value” van een resource en “eigenschap van een resource”: dit doe ik via een demo: zoek in Google op “AWS CloudFormation S3 bucket”, laat zien dat AWS::S3::Bucket het eerste resultaat is, klik in het rechter menu op “Return values” en laat zien dat !Ref de bucket name terug geeft en !GetAtt bijvoorbeeld de ARN, of de DomainName, of … terug geeft.
* De URL’s zijn bedoeld voor als je meer wilt weten, binnen deze workshop heb je aan de functies op deze slide voldoende.

**Slide 10:**

* Maakt gebruik van de Systems Manager (SSM) Parameter Store, als je daar nooit mee gewerkt hebt: knip en plak gewoon deze code (ook in de handouts/in de gitlab repository) en maak er gebruikt van. Heb je ooit wel van SSM Parameters gebruik gemaakt dan herken je bij Default de naam van de SSM parameter. Het type is AWS::EC2::Image::Id, dit zorgt er voor dat je altijd een geldige, bestaand Image Id terug krijgt.
* Het voordeel hiervan is dat je dus nooit meer ID’s hoeft terug te zoeken in de GUI of op websites, maar dat je altijd de meest recente Amazon Linux 2 AMI terug krijgt.
* AWS heeft meer van dit soort standaard SSM parameters die altijd bestaan, google op **/aws/service/ami-amazon-linux-latest/amzn2-ami-hvm-x86\_64-ebs** om ze te vinden.
* Het gebruik is hetzelfde als elke andere parameter: met !Ref AmazonLinux2AMI (dus: !Ref en de door jou gegeven naam van de parameter)

**Slide 13:**

* Ervaren CloudFormation gebruikers maken eerst een diagram (bijv. via draw io) en gaan dan pas code schrijven.
* Route table: let op, 172.16.0.0 etc. zijn IP ranges die je -mag- gebruiken, maar je mag ook wat anders gebruiken, bijvoorbeeld 10.0.0.0 of 192.168.0.0. In het standaard AWS icon staat ook in die gevallen nog steeds 172.16.0.0, 172.16.1.0 en 172.16.2.0.
* De handout is opgebouwd dat bij elke opdracht eerst het diagram staat, daarna de hints en daarna de volledige oplossing. Je vindt de volledige oplossing ook in de github repository.

**Slide 15:**

* De parameters die hier genoemd staan en ook de userdata is beschikbaar via zowel de handout als de github repository.
* Je maakt gebruik van een MySQL database die elders draait (zie evt. in de folder voorbereidingen), user-id, wachtwoord en endpoint staan op een volgende slide

**Slide 16:**

* Het is mooier als we de database zelf deployen
* **LET OP:** normaal gesproken maak je de VPC, de database en de Virtual Machine in drie verschillende stacks. De reden hiervoor is dat als iets in de Virtual Machine stack wijzigt, je eventueel de stack kunt weggooien en opnieuw opbouwen zonder dat je de database weggooit en opnieuw moet aanmaken.

**Slide 18:**

In de demo laat ik zien hoe je een CloudFormation template voor een VPC kunt maken. Je ziet het resultaat in de Gitlab repository (file opdracht1.yaml). Je begint met AWSTemplateFormatVersion en Description. Geef voor het vervolg ook je eigen naam als parameter mee:

# opdracht1.yaml

# ==============

AWSTemplateFormatVersion: "2010-09-09"

Description: "Opdracht 1"

Parameters:

  Name:

    Type: String

    Default: frederique

Resources:

We zullen samen de eerste resource gaan doen: de VPC.

* Zoek in Google naar AWS CloudFormation VPC. De eerste hit is:

Text

Description automatically generated

* Dit is direct de goede. Klik hierop. Je ziet nu heel veel parameters:

Text

Description automatically generated with medium confidence

* Je kunt op elk van deze properties klikken. Advies: ga eerst in de GUI zelf een VPC aanmaken en kijk welke properties je wilt veranderen. In het algemeen zijn dit ook de enige parameters die je mee hoeft te geven.

Graphical user interface, text, application

Description automatically generated

* Je ziet in de GUI dat je een name tag mee wilt geven en een IPV4 Cidr block, bijvoorbeeld 10.0.0.0/16. Je ziet ook de name tag.
* In de documentatie zie je dat CidrBlock een string is. Onder Tags zie je een list van het type tag. Als je op de link “tag” klikt, dan zie je dat iedere tag bestaat uit een Key (string) en een value (ook een string)
* Het resultaat onder Resources: is dan als volgt:

  WordPressVpc:

    Type: AWS::EC2::VPC

    Properties:

      CidrBlock: "10.0.0.0/16"

      Tags:

        - Key: Name

          Value: "wordpressvpc-frederique"

* Je kunt de name tag in je CloudFormation template natuurlijk als letterlijke tekst meegeven, zoals in het voorbeeld hierboven. Veel netter is het om frederique af te leiden uit de parameter. Het volledige voorbeeld wordt dan:

# opdracht1.yaml

# ==============

AWSTemplateFormatVersion: "2010-09-09"

Description: "Opdracht 1"

Parameters:

  Name:

    Type: String

    Default: frederique

Resources:

  WordPressVpc:

    Type: AWS::EC2::VPC

    Properties:

      CidrBlock: 10.0.0.0/16

      Tags:

        - Key: Name

          Value: !Sub "wordpressvpc-${Name}"