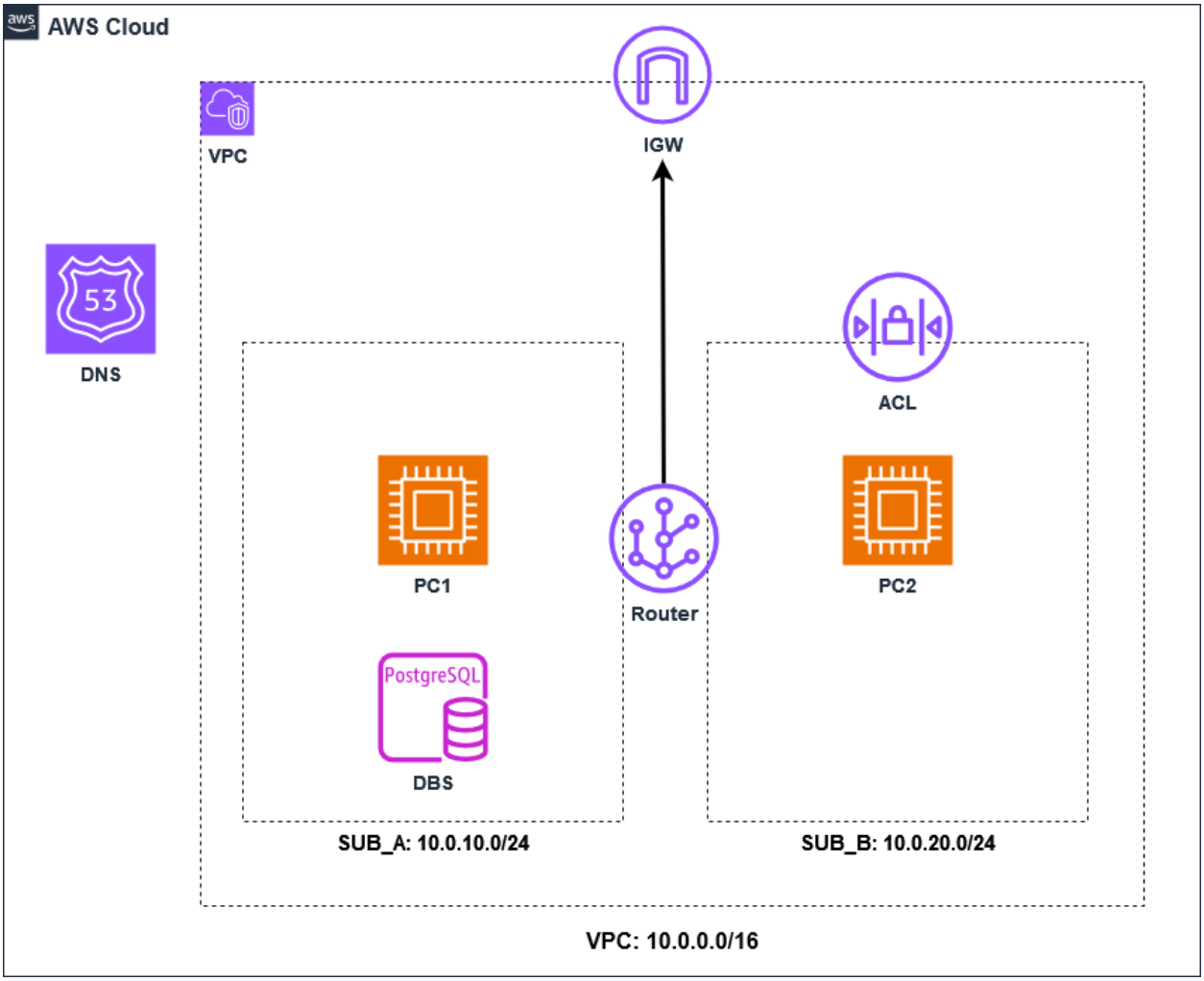
Tvorba sieťových prostredí v AWS cloud-e

Obsahom Lab manuálu je ukázať prácu so základnými službami v AWS cloude potrebnými pre prácu a vytváranie sieťových prostredí.

Topológia

Práca so základnými službami pre interakciu so sieťovým prostredím bude demonštrovaná na nasledujúcej topológii:

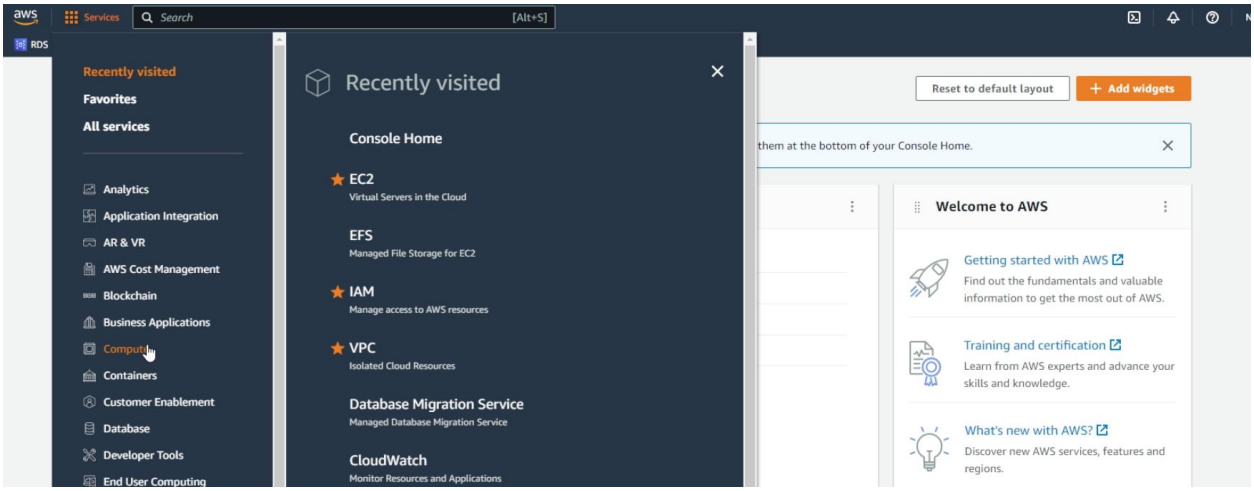


Úlohy

Nasledujúce úlohy ukážu, ako vytvoriť VPC sieť, ako túto sieť rozdeliť na menšie podsiete, ako pracovať so smerovaním, ako umožniť prístup zariadeniam vo VPC sieti do internetu, ako zabezpečiť prístup ku konkrétnym zdrojom, ako vytvoriť virtuálny server a pripojiť sa na neho využitím protokolov SSH a HTTP, ako povoliť komunikáciu medzi dvojicou vnútorných zdrojov, ako umožniť ich komunikáciu využitím doménových názvov, ako vytvoriť databázu a prepojiť ju s virtuálnym serverom a ako obmedziť komunikáciu medzi podsieťami. Poslednou znázornenou úlohou je ukážka ako vybrané vyššie uvedené koncepty automatizovať využitím IaaC nástroja CloudFormation.

Pre prácu s AWS konzolou je potrebné mať vlastný účet. V tomto prípade je potrebné dať pozor na to, že niektoré AWS služby sú spoplatnené a ich použitie môže byť účtované. Druhou možnosťou je zakúpenie kurzu, napr. cez platformu acloudguru, v rámci ktorej sú k dispozícii 60 - 90 minútové cvičenia, ktorých súčasťou je prístup do AWS konzoly.

Všetky vyššie uvedené funkcionality sú v AWS cloude reprezentované na to určenou službou, ktorú je možné nájsť využitím vyhľadávacieho okna v AWS konzole.



Virtual Private Cloud (VPC)

Vytvorenie virtuálnej privátnej siete je možné využitím služby VPC, v rámci ktorej stačí zvoliť možnosť **Create VPC** a definovať nasledujúce parametre:

- Typ: **typ VPC siete**
- Tag: **označenie VPC siete**
- IPv4 CIDR block: **spôsob definovania adresného rozsahu**
- IPv4 CIDR: **10.0.0.0/16 (samotný adresný rozsah)**
- IPv6: **No IPv6 CIDR Block**
- Tenancy: **Default**

Resources to create Info

Create only the VPC resource or the VPC and other networking resources.

VPC only | VPC and more

Name tag - *optional*

Creates a tag with a key of 'Name' and a value that you specify.

VPC

IPv4 CIDR block Info

IPv4 CIDR manual input

IPAM-allocated IPv4 CIDR block

IPv4 CIDR

10.0.0.0/16

IPv6 CIDR block Info

No IPv6 CIDR block

IPAM-allocated IPv6 CIDR block

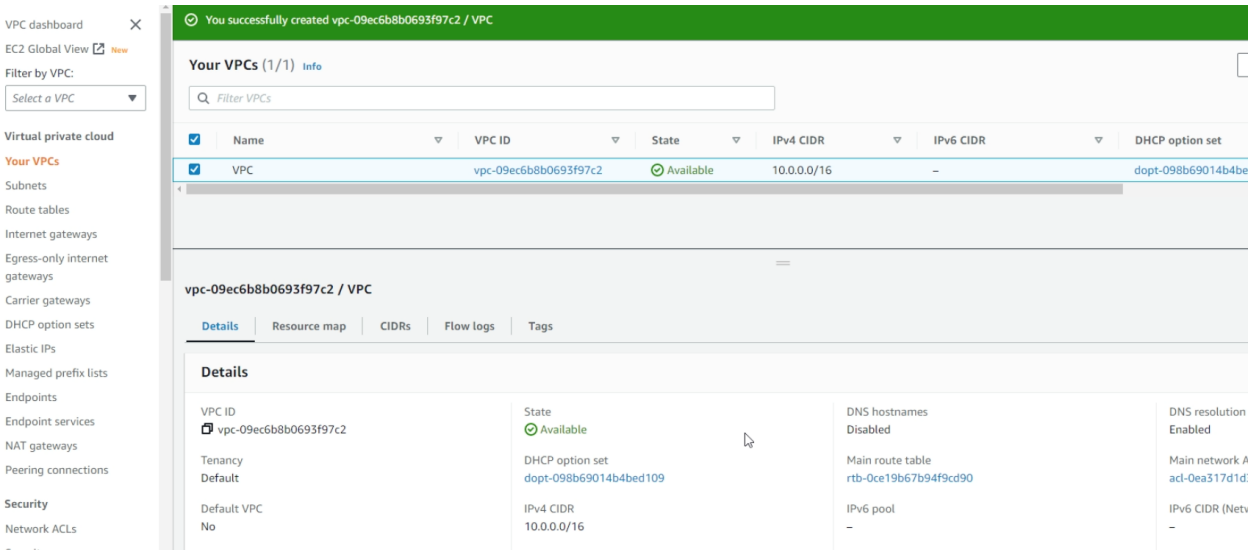
Amazon-provided IPv6 CIDR block

IPv6 CIDR owned by me

Tenancy Info

Default

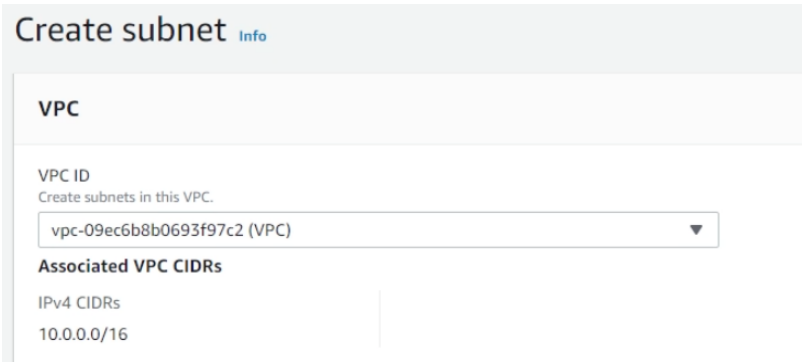
Tenancy slúži na určenie toho, či zdroje v našej VPC sieti budú spustené na zdieľaných serveroch (default - lacnejšie riešenie), alebo na dedikovaných (dedicated - drahšie riešenie).

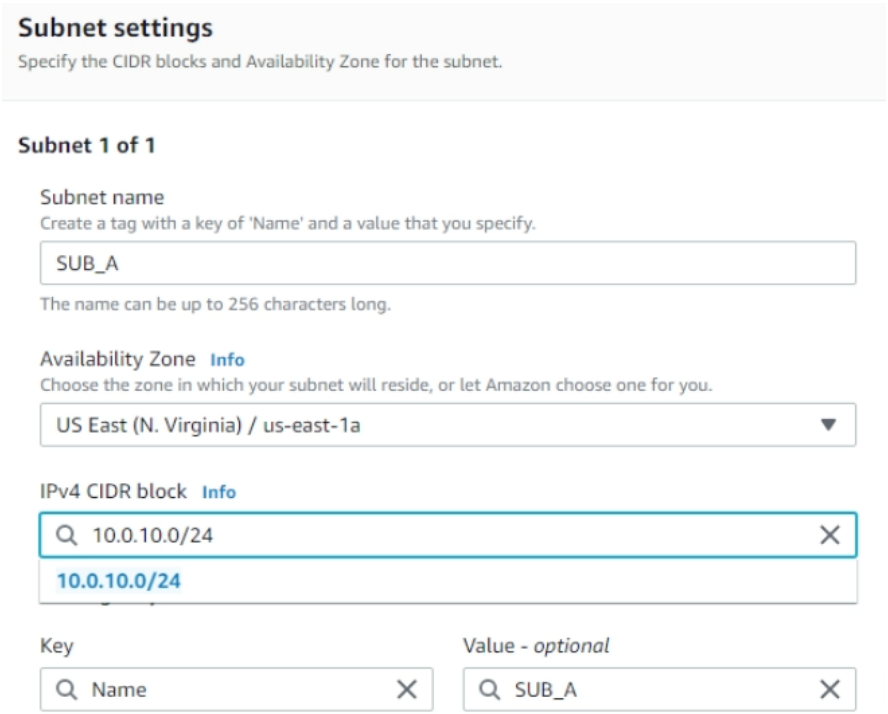


Subnets

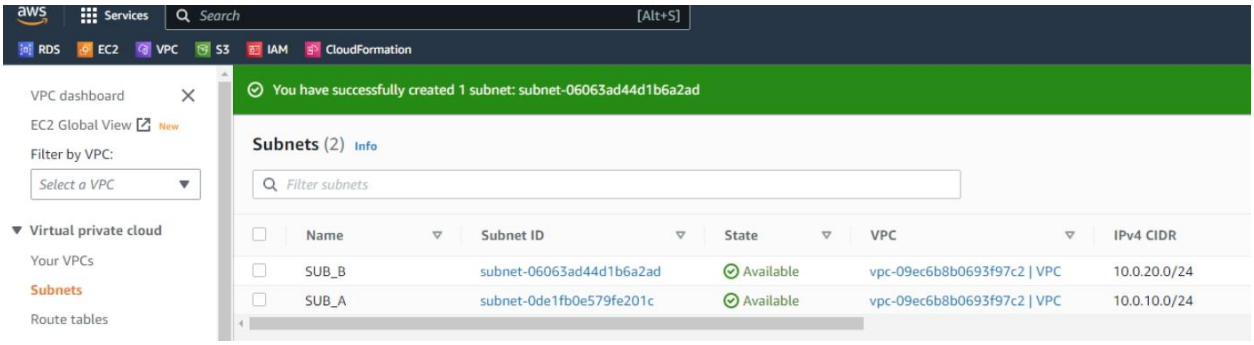
Definovanie podsiete vyžaduje nastavenie nasledujúcich parametrov:

- VPC ID: **id vytvorenej VPC, do ktorej bude podsieť patriť**
- Subnet name: **meno podsiete**
- Availability Zone: **AZ, v ktorej sa bude podsieť nachádzať**
- IPv4 CIDR block: **adresný rozsah danej podsiete**



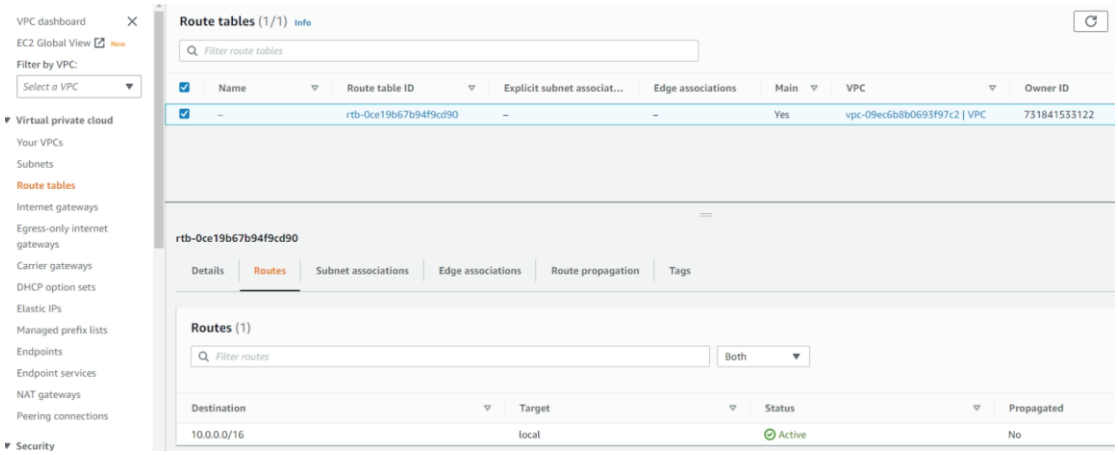


Rovnakým spôsobom vytvorte aj podsieť SUB_B, ktorej prideľte adresný rozsah 10.0.20.0/24.



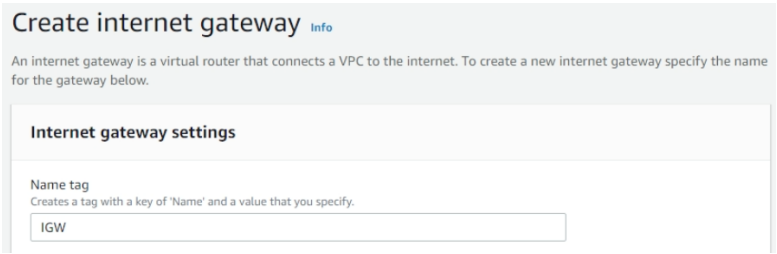
Routing table concepts

V AWS existuje globálna smerovacia tabuľka, v ktorej sa štandardne nachádzajú všetky podsiete a má súhrný záznam na dosiahnutie celého adresného rozsahu celej VPC. Pre väčšiu granularitu je ale možné vytvoriť lokálne smerovacie tabuľky a v nich definovať, kam daná podsieť dokáže smerovať dáta. Podsieť, v ktorej sa nachádza default route sa označuje ako verejná.



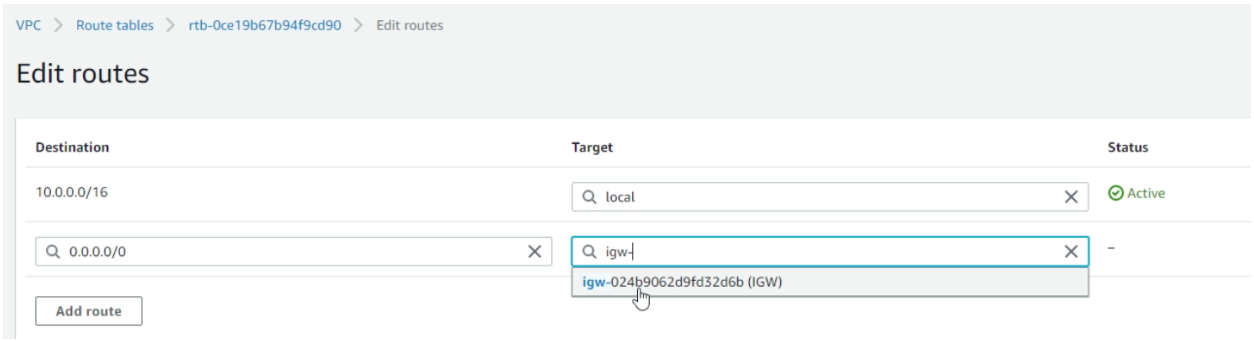
Internet Gateway

IGW je služba umožňujúca prístup k internetu. Táto služba môže byť označená ako smer pri definovaní default route. Pri vytváraní IGW stačí definovať jej názov.



Vytvorenú IGW je následne potrebné priradiť (Attach) k existujúcej VPC cez záložku Actions - Attach to VPC, kde stačí definovať cieľoví VPC sieť.

Do smerovacej tabuľky následne stačí pridať default route a ako cieľ nastaviť vytvorenú IGW.



Elastic Compute Cloud (EC2)

EC2 predstavuje inštanciu virtuálneho zariadenia. Pri jej vytváraní je potrebné definovať nasledujúce parametre:

- Priradiť do VPC a podsiete
- Zvoliť OS (AWS linux) a typ inštancie
- Vygenerovať SSH kľúč pre vzdialenú správu (.pem file)
- Nastaviť sieťové parametre a povoliť priradenie verejnej IP (potrebné pre umožnenie dostupnosti z internetu)
- Vytvoriť SG a povoliť v nej SSH (vzdialený prístup) a HTTP (prístup na webovú stránku)
- Vložiť script na inštaláciu Web servera

Konfigurácia vyššie uvedeného sa začína v záložke EC2 - Launch an instance:

Launch an instance Info

Amazon EC2 allows you to create virtual machines, or instances, that run on the AWS Cloud. Quickly get started by following the simple steps below.

Name and tags Info

Name

PC1

Add additional tags

Application and OS Images (Amazon Machine Image) Info

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. Search or Browse for AMIs if you don't see what you are looking for below

Search our full catalog including 1000s of application and OS images

Quick Start

Amazon Linux | macOS | Ubuntu | Windows | Red Hat | S

Browse more AMIs

Including AMIs from AWS, Marketplace and the Community

Amazon Machine Image (AMI)

Amazon Linux 2 AMI (HVM) - Kernel 5.10, SSD Volume Type — Free tier eligible

ami-0aa7d40eeae50c9a9 (64-bit (x86)) / ami-084237e82d7842286 (64-bit (Arm))

Virtualization: hvm ENA enabled: true Root device type: ebs

Instance type Info

Instance type

t2.micro — Free tier eligible

Family: t2 1 vCPU 1 GiB Memory

On-Demand Linux pricing: 0.0116 USD per Hour

On-Demand Windows pricing: 0.0162 USD per Hour

Compare instance types

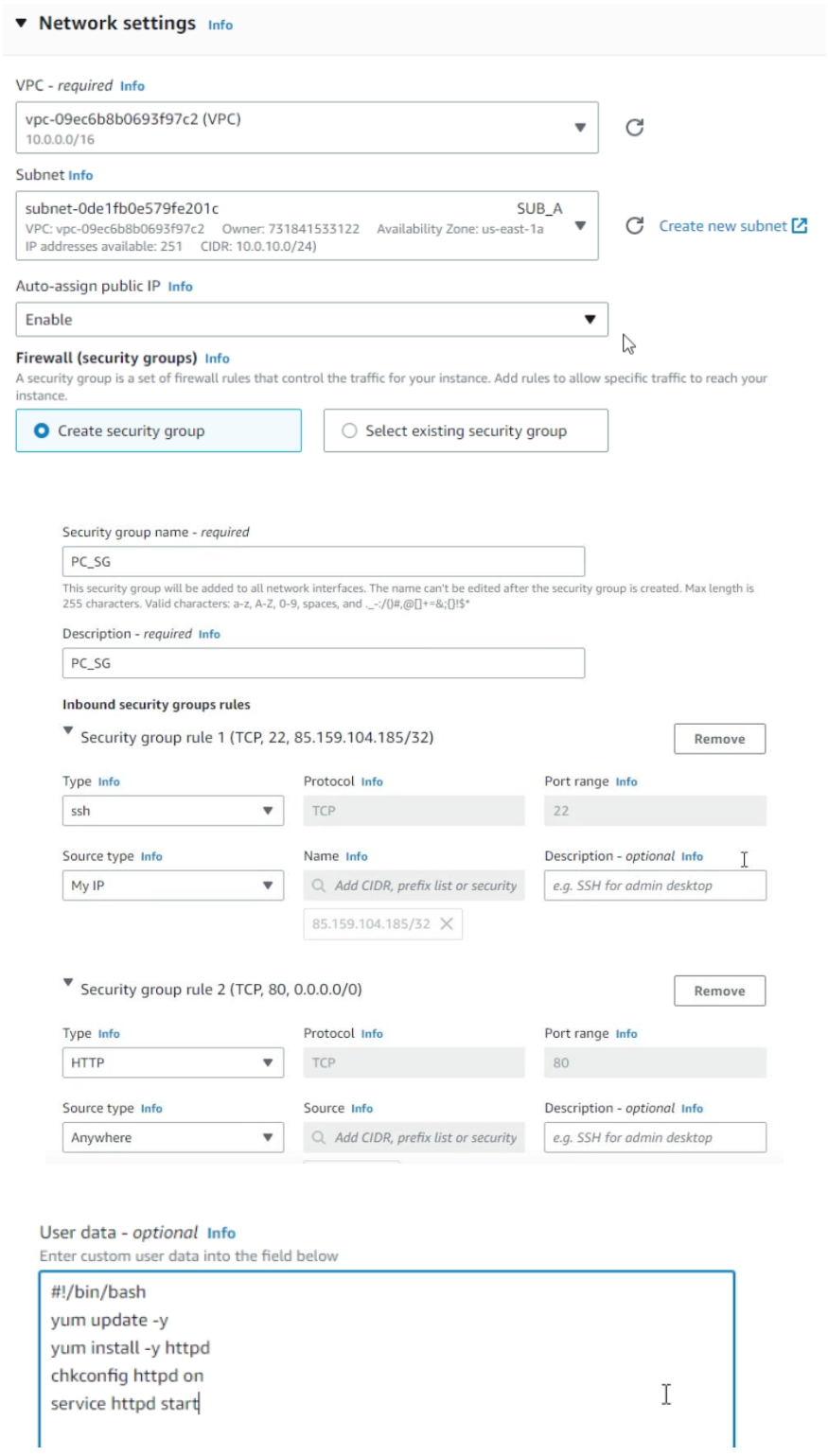
Key pair (login) Info

You can use a key pair to securely connect to your instance. Ensure that you have access to the selected key pair before you launch the instance.

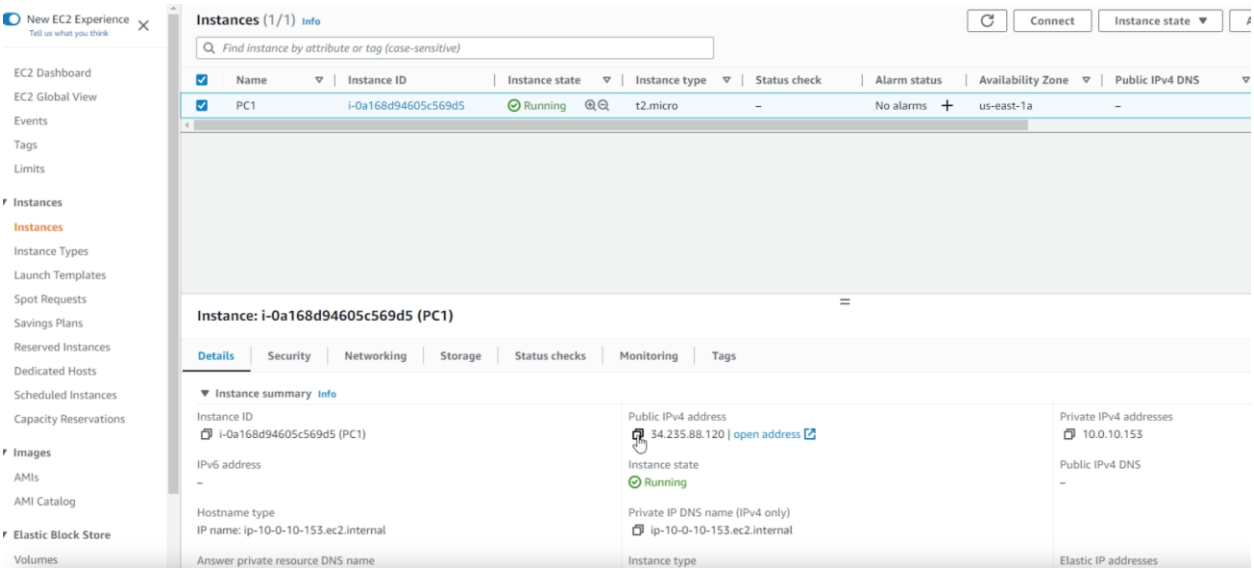
Key pair name - *required*

key

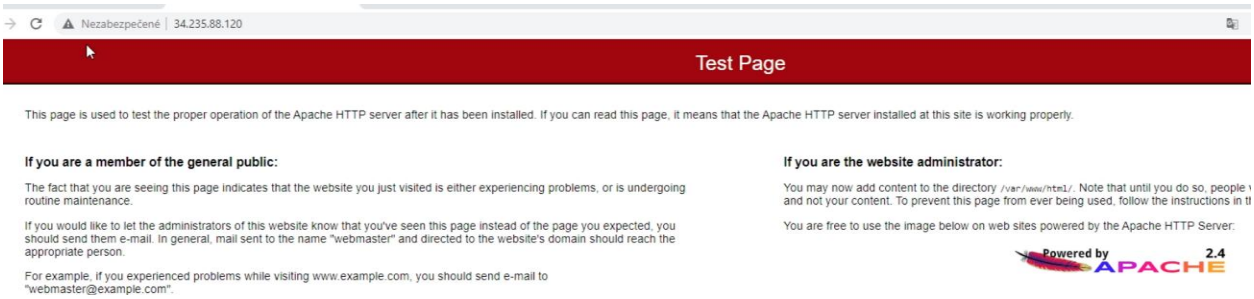
Create new key pair



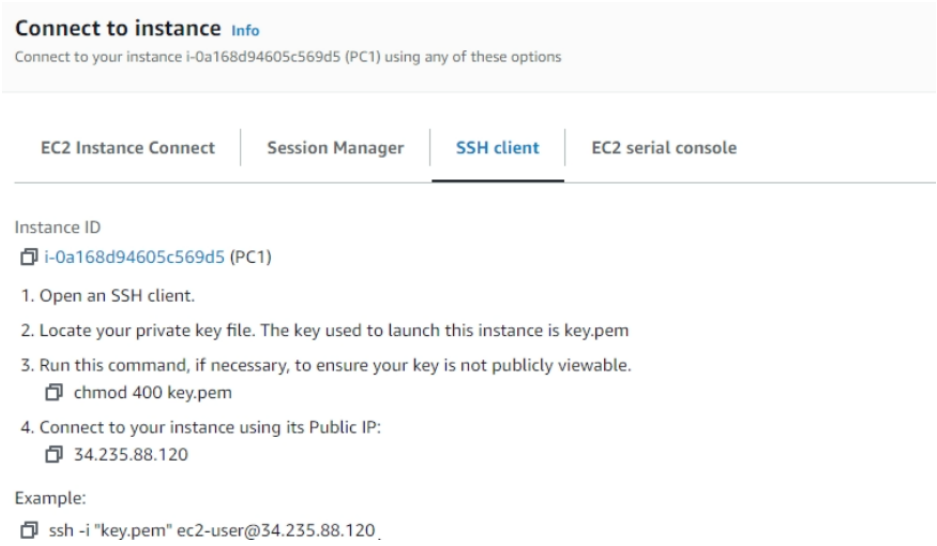
Zvyšok parametrov môže ostať v štandardnom nastavení.



Overiť úspešnosť riešenie je možné zobrazením webovej stránky dostupnej na verejnej adrese servera.



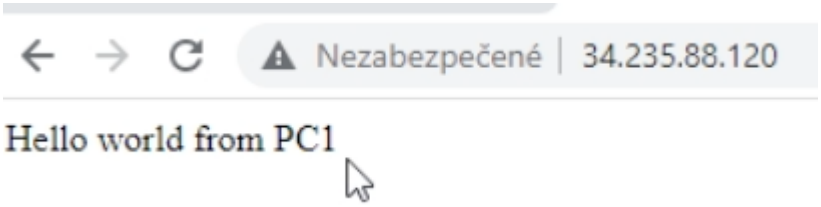
Overenie pripojenia na SSH daného servera je možné cez záložku **Connect**, kde je možné nájsť postupnosť krokov pre ssh pripojenie (vyžaduje existenciu .pem súboru definovaného pri vytváraní inštancie).



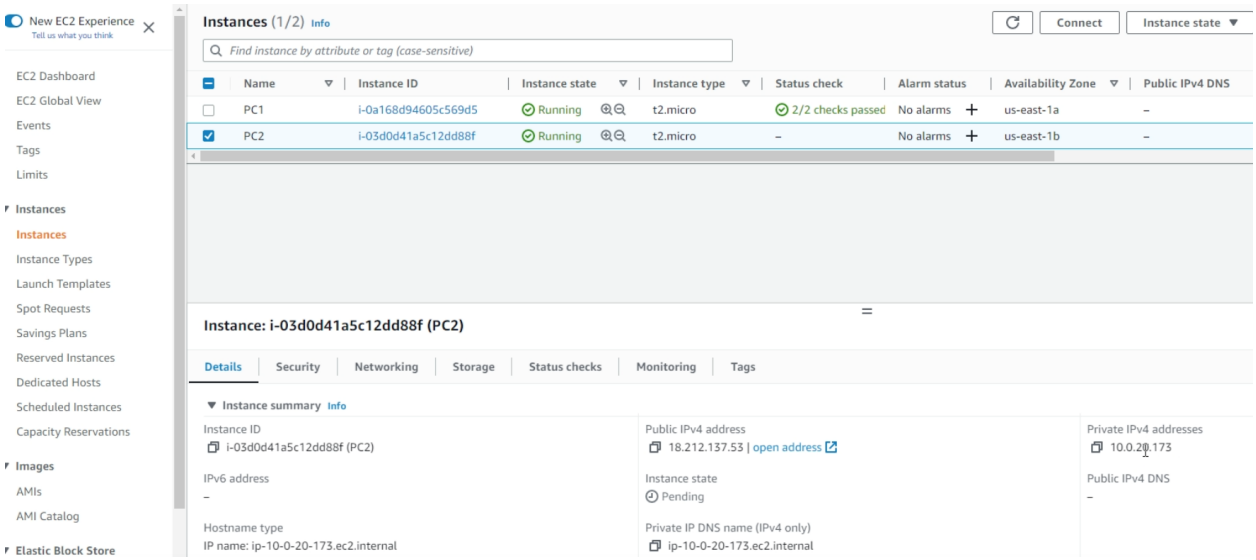
```
C:\Users\rp385yq\Desktop>ssh -i "key.pem" ec2-user@34.235.88.120
The authenticity of host '34.235.88.120 (34.235.88.120)' can't be established.
ECDSA key fingerprint is SHA256:KDSEkv22ImcndfwCK9NcxaZWPoqlWQigCjN86Coqzf0.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '34.235.88.120' (ECDSA) to the list of known hosts.

       __|  __|_  )
       _|  (     /   Amazon Linux 2 AMI
      ___|\___|___|

https://aws.amazon.com/amazon-linux-2/
[ec2-user@ip-10-0-10-153 ~]$
```

Vytvorením súboru **/var/www/html/index.htm**, je možné upraviť obsah zobrazovanej webovej stránky.

```
<html>
 Hello world from PC1
</html>
```

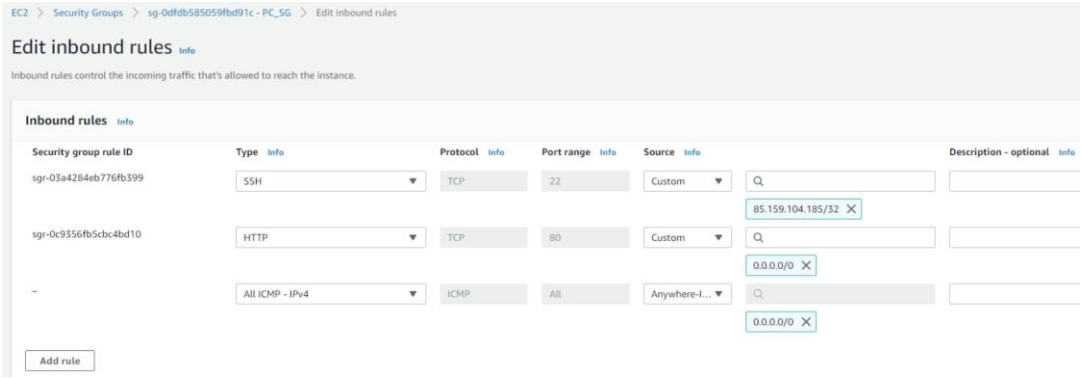


Rovnakým spôsobom je potrebné vytvoriť aj zariadenie PC2.



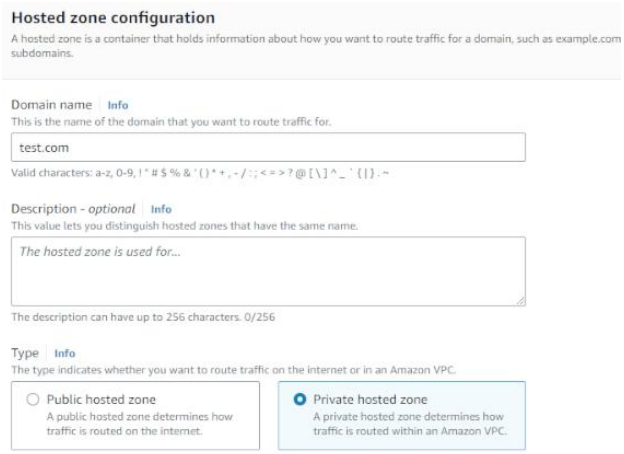
Security Group (SG)

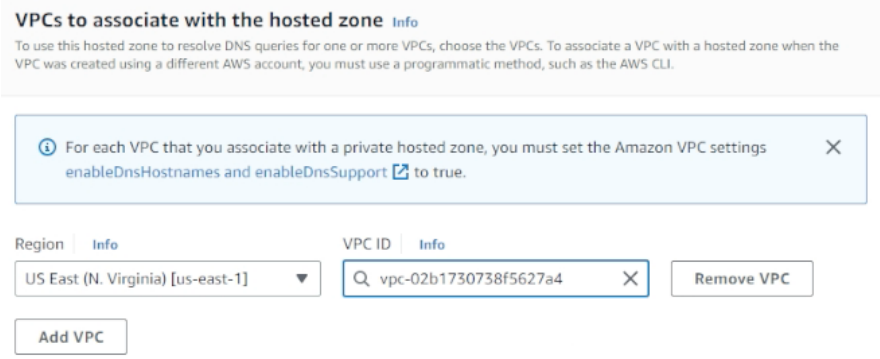
Bezpečnostné skupiny slúžia na ochranu konkrétnych sieťových kariet používaných zdrojov. Obsahujú len povolenia a môžu byť definované v smere do ale aj v smere z daného zdroja. Pravidlá majú stateful charakter a teda všetko čo je v jednom smere povolené SG, tak automaticky aj odpovede na túto komunikáciu sú povolené. Štandardne je všetka komunikácia smerom von povolená, avšak smerom dnu len to, čo sme definovali pri vytváraní EC2 inštancie. Kvôli tomu, aktuálne nie je možná komunikácia protokolom ICMP z PC1 na PC2, pretože PC2 v smere in nepovoľuje ICMP správy. Opraviť tento nedostatok je možné pridaním ďalšieho záznamu do SG.



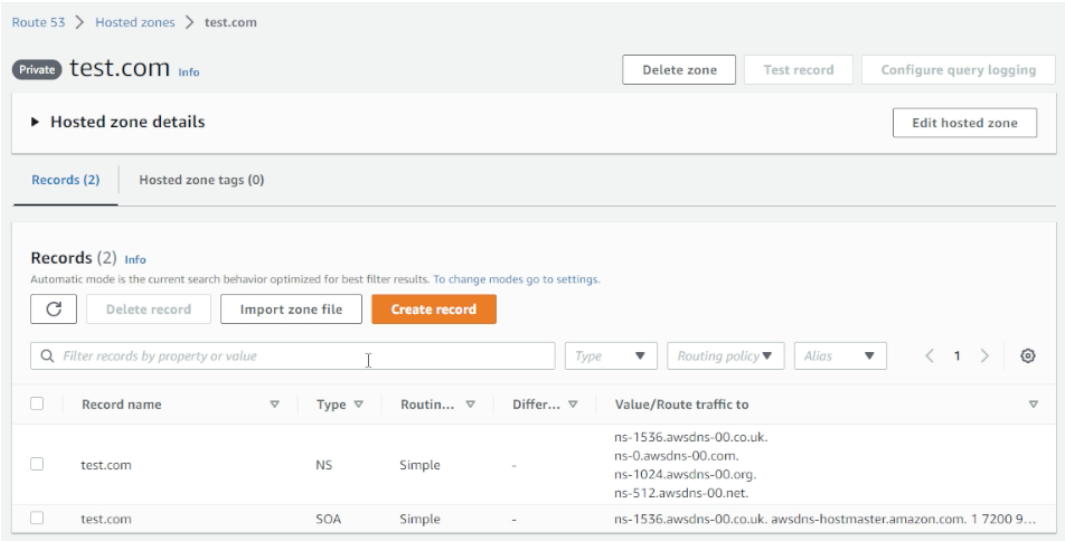
Route 53

Route 53 je služba pre mapovanie IP adries na doménové názvy. Je to AWS DNS služba. Štandardne v rámci AWS funguje lokálne DNS, ktoré umožňuje dosiahnuteľnosť zdrojov využitím ich **Private IP DNS name**. Pre definovanie vlastných doménových názvov je ale potrebné dodatočné nastavenie. V rámci služby Route 53 je potrebné vytvoriť **Private Hosted zone**. Existuje aj Public hosted zone, no tá by spravovala globálne dostupné DNS záznamy, avšak vyžaduje zakúpenie domény.
Vytvorenie Privátnej hosted zóny vyžaduje nastavenie jej typu, názvu domény a VPC siete.

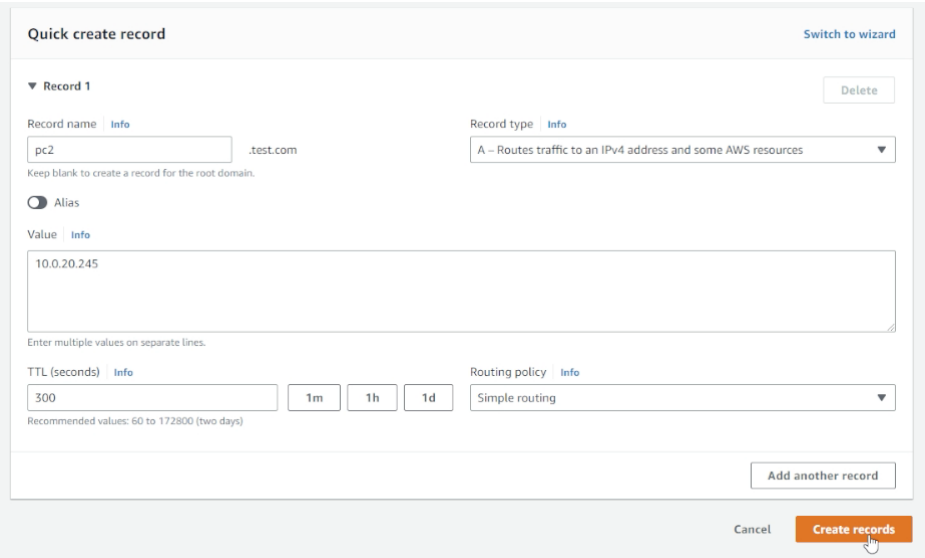




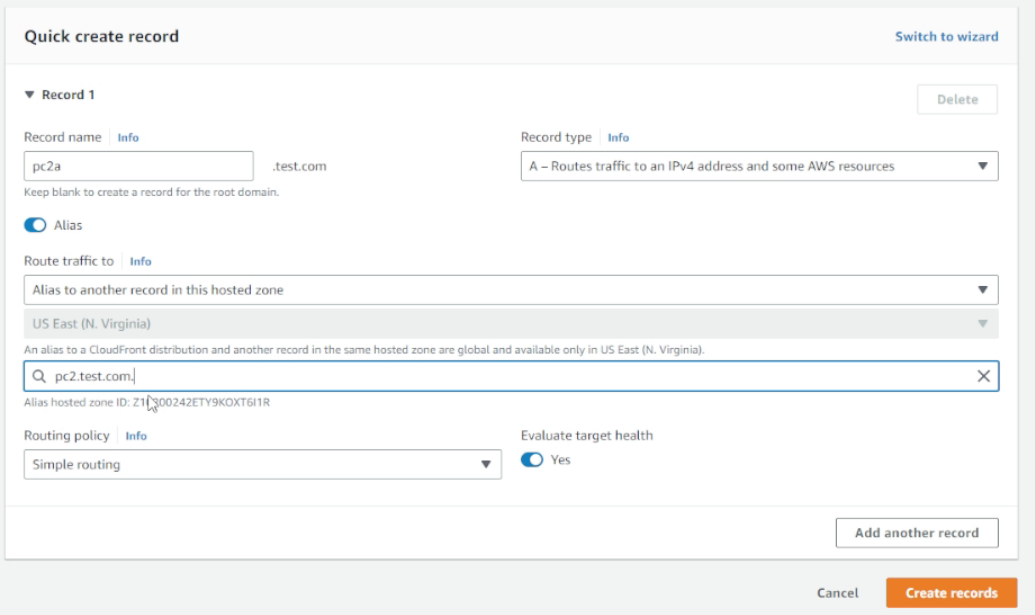
Po vytvorení hosted zóny sa vytvorí základná štruktúru DNS záznamu definujúca NS a SOA atribúty.

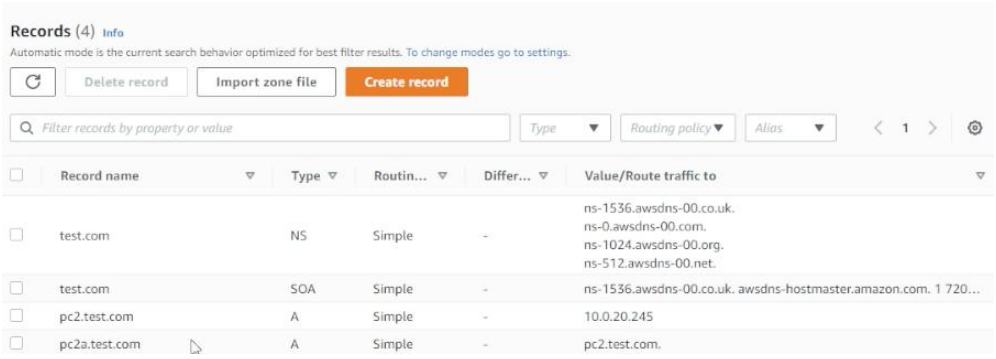


Teraz už len stačí definovať **A** záznam mapujúci požadovaný názov domény, napr. Pc2.test.com na IPv4 adresu daného zariadenia.



Funkciou Route 53 je tvorba tzv. Alias záznamov, ktoré umožňujú efektívne mapovanie doménového názvu na doménový názov.

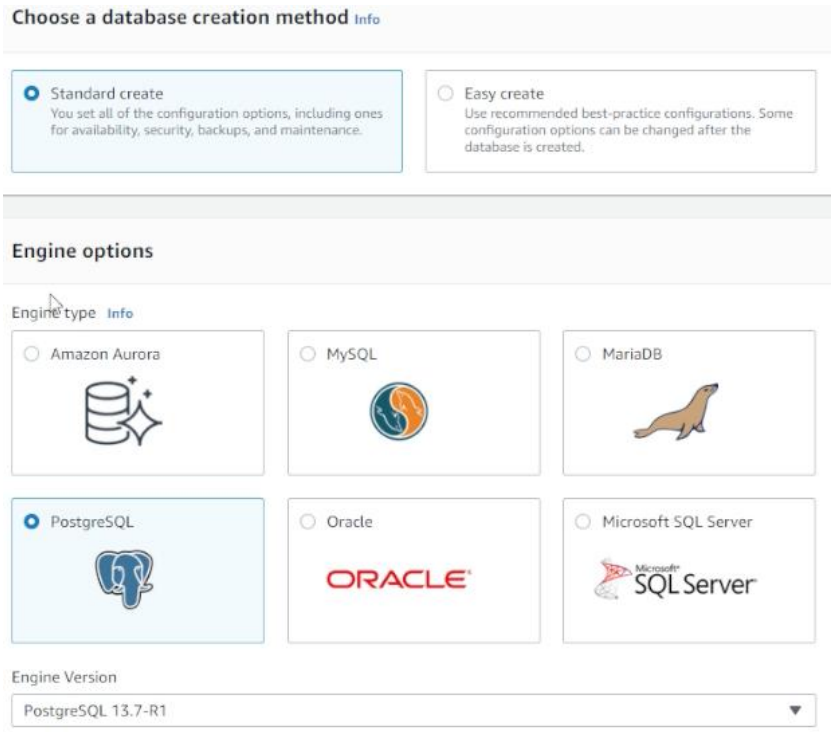




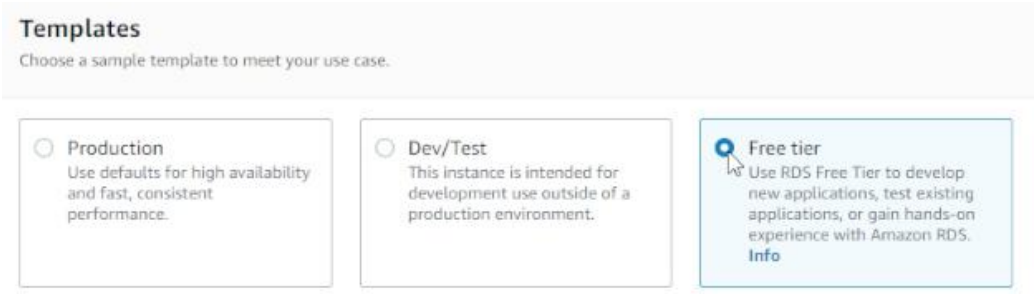

```
[ec2-user@ip-10-0-10-84 ~]$ ping pc2a.test.com
PING pc2a.test.com (10.0.20.245) 56(84) bytes of data.
64 bytes from ip-10-0-20-245.ec2.internal (10.0.20.245): icmp_seq=1 ttl=
255 time=0.666 ms
64 bytes from ip-10-0-20-245.ec2.internal (10.0.20.245): icmp_seq=2 ttl=
255 time=0.735 ms
```

Rational Database Service (RDS)

RDS je službou pre vytváranie relačných databáz. Databázu je možné vytvoriť kliknutím na tlačidlo **Create database** a voľbou konkrétneho typu databázy.



Ako template je potrebné zvoliť Free tier.



Ďalej je potrebné nastaviť **Meno databázy** a prihlasovacie údaje (**Master username** a **Master password**). Zvyšok parametrov je možné nechať bez zmeny. Jediným parametrom, ktorý je ešte potrebné nastaviť je **Initial database name**, ktorý je možné nájsť v časti Additional Configuration. Tiež je potrebné definovať v akej VPC sieti a AZ sa databáza bude nachádzať.

Settings

DB instance identifier Info

Type a name for your DB instance. The name must be unique across all DB instances owned by your AWS account in the current AWS Region.

database-1

The DB instance identifier is case-insensitive, but is stored as all lowercase (as in "mydbinstance"). Constraints: 1 to 60 alphanumeric characters or hyphens. First character must be a letter. Can't contain two consecutive hyphens. Can't end with a hyphen.

▼ Credentials Settings

Master username Info

Type a login ID for the master user of your DB instance.

postgres

1 to 16 alphanumeric characters. First character must be a letter.

☐ Manage master credentials in AWS Secrets Manager - *new*

Manage master user credentials in Secrets Manager. RDS can generate a password for you and manage it throughout its lifecycle.

☐ Auto generate a password

Amazon RDS can generate a password for you, or you can specify your own password.

Master password Info

Constraints: At least 8 printable ASCII characters. Can't contain any of the following: / (slash), '(single quote), "(double quote) and @ (at sign).

Confirm master password Info

▼ Additional configuration

Database options, encryption turned on, backup turned on, backtrack turned off, maintenance, CloudWatch Logs, delete protection turned off.

Database options

Initial database name Info

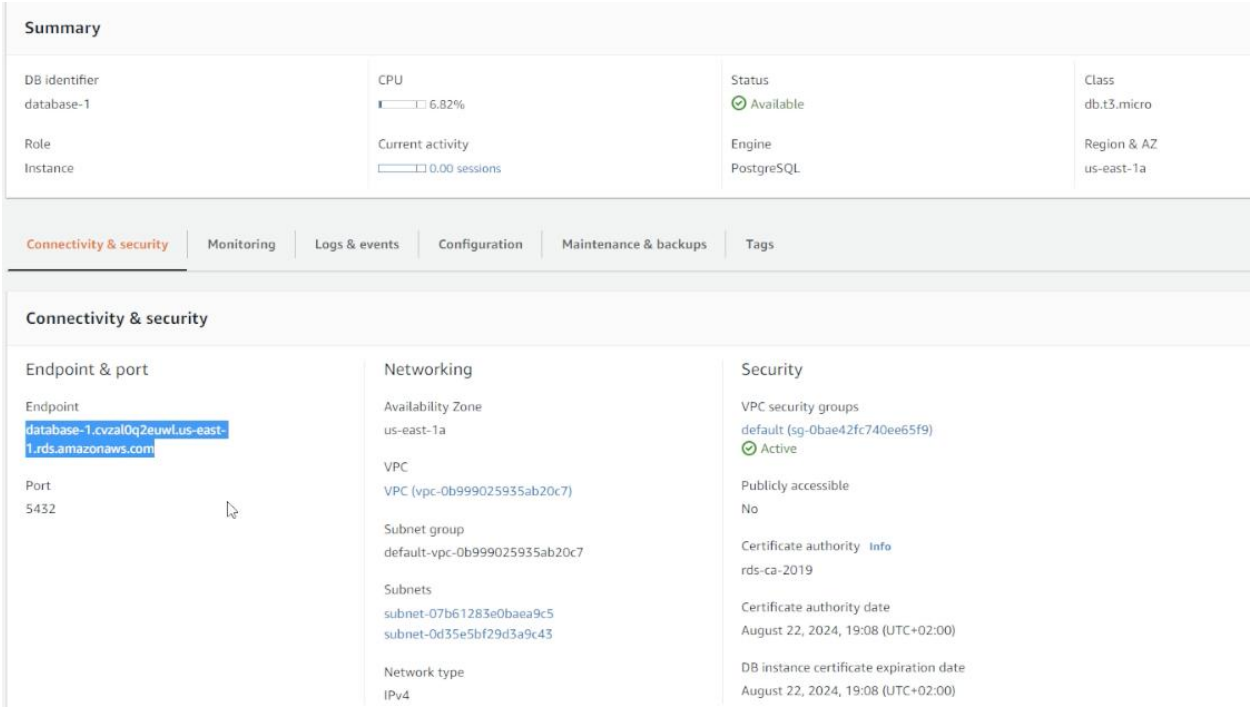
myDB

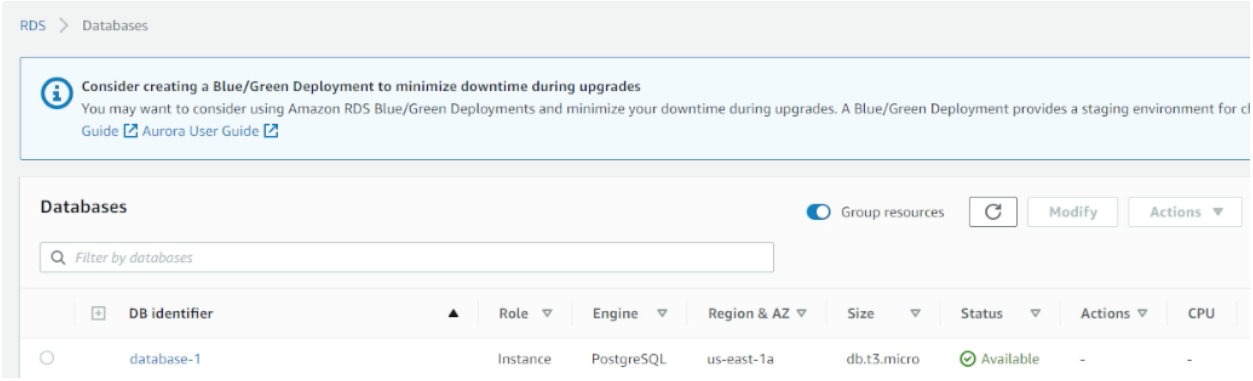
If you do not specify a database name, Amazon RDS does not create a database.

DB parameter group Info

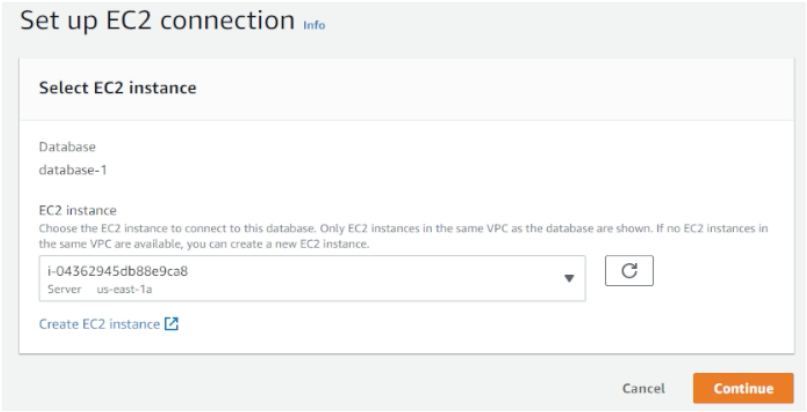
default.postgres13 ▼

Po dlhšej chvíli dôjde k vytvoreniu databázy.

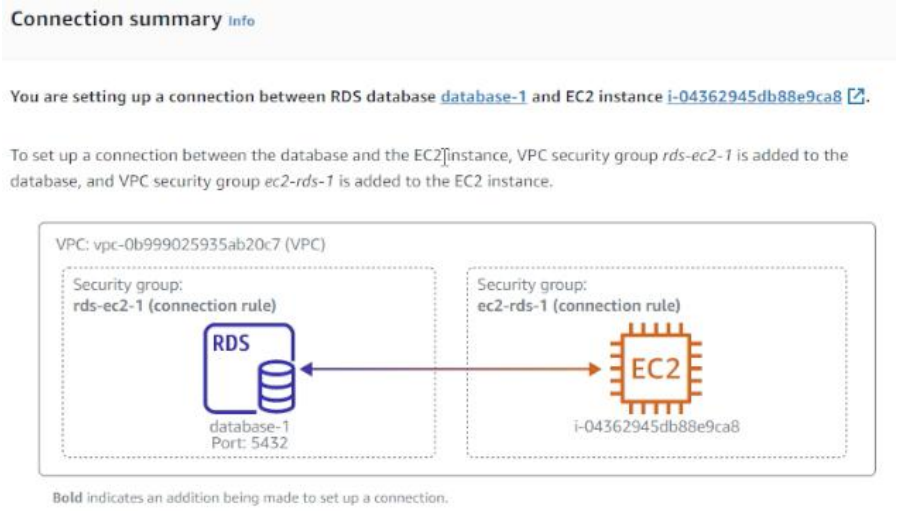


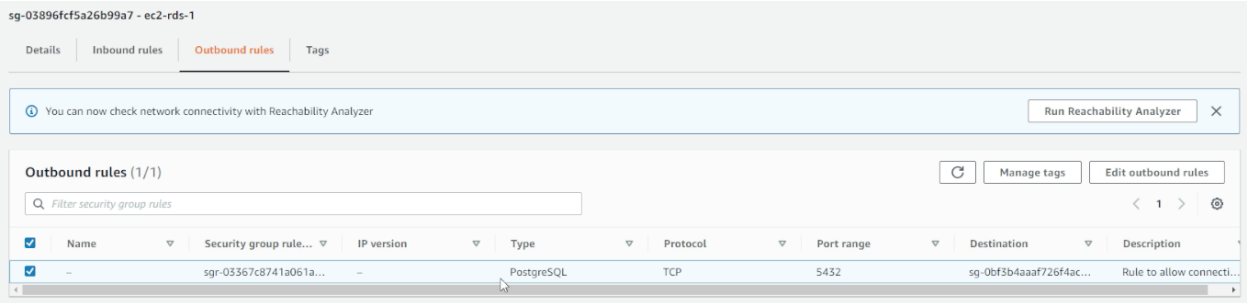


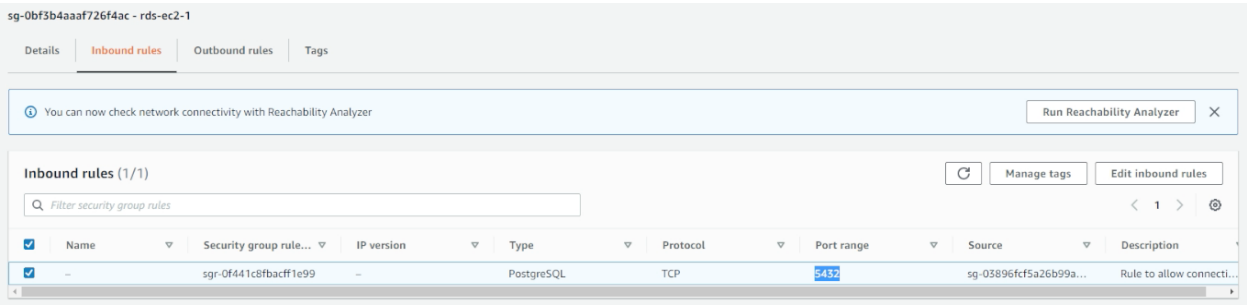
Teraz je potrebné v časti vytvorenej databázy kliknúť na záložku **Actions** a zvoliť možnosť **Set up EC2 connection**. Tu stačí zvoliť EC2 inštanciu, s ktorou je potrebné vytvoriť spojenie.



Vytvorené nastavenie automaticky vytvorí potrebné SG a priradí ich na inštancie EC2 a DB.







Na pripojenie sa z EC2 inštancie na databázu je potrebné spustiť nasledujúce príkazy z EC2 inštancie (inštalovanie db klienta a vykonanie prihlásenia):

```
sudo yum install postgresql15

psql -h <IP_Address> -p <port_no> -d <database_name> -U <DB_username> -W
psql -h database-1.cvzal0q2euwl.us-east-1.rds.amazonaws.com -p 5432 -d myDB -U postgres -W
```

```
[ec2-user@ip-10-0-10-199 ~]$ psql -h database-1.cvzal0q2euwl.us-east-1.rds.amazonaws.com -p
5432 -d myDB -U postgres -W
Password for user postgres:
psql (9.2.24, server 13.7)
WARNING: psql version 9.2, server version 13.0.
         Some psql features might not work.
SSL connection (cipher: ECDHE-RSA-AES256-GCM-SHA384, bits: 256)
Type "help" for help.

myDB=>
```

Interakcia s Postgresql databázou je následne vykonateľná využitím bežných príkazov:

```
\l - zoznam databáz
CREATE TABLE students(
  user_id serial PRIMARY KEY,
  username VARCHAR(50),
  email VARCHAR(50)
);

\dt - zobrazenie vytvorených tabuliek

INSERT INTO students(username, email)
VALUES ('student1', 'student1@mail.com');

SELECT * from students;

\q
```

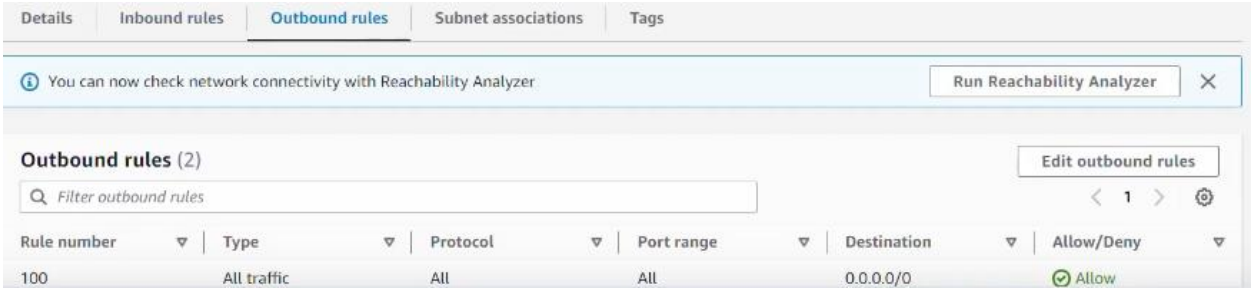
```
myDB=> \l
                                  List of databases
   Name    |  Owner   | Encoding |   Collate   |    Ctype    |   Access privileges
-----------+----------+----------+-------------+-------------+-----------------------
 myDB      | postgres | UTF8     | en_US.UTF-8 | en_US.UTF-8 |
 postgres  | postgres | UTF8     | en_US.UTF-8 | en_US.UTF-8 |
 rdsadmin  | rdsadmin | UTF8     | en_US.UTF-8 | en_US.UTF-8 | rdsadmin=CTc/rdsadmin+
           |          |          |             |             | rdstopmgr=Tc/rdsadmin
 template0 | rdsadmin | UTF8     | en_US.UTF-8 | en_US.UTF-8 | =c/rdsadmin          +
           |          |          |             |             | rdsadmin=CTc/rdsadmin
 template1 | postgres | UTF8     | en_US.UTF-8 | en_US.UTF-8 | =c/postgres          +
           |          |          |             |             | postgres=CTc/postgres
(5 rows)
```

```
myDB=> CREATE TABLE students(
myDB(>   user_id serial PRIMARY KEY,
myDB(>   username VARCHAR(50),
myDB(>   email VARCHAR(50)
myDB(> );
CREATE TABLE
myDB=>
myDB=>
myDB=>
myDB=> \dt
          List of relations
 Schema |   Name   | Type  |  Owner
--------+----------+-------+----------
 public | students | table | postgres
(1 row)
```

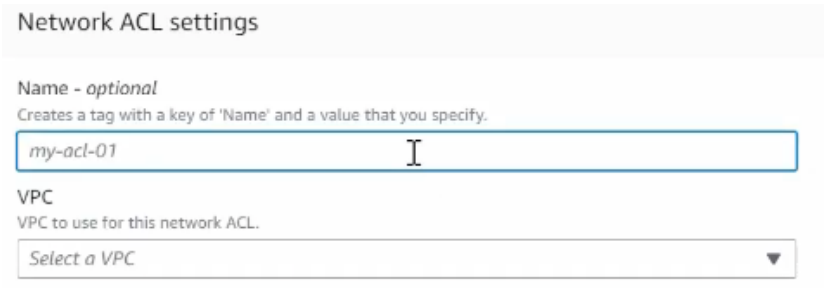
```
myDB=> INSERT INTO students(username, email)
myDB-> VALUES ('student1', 'student1@mail.com');
INSERT 0 1
myDB=>
myDB=>
myDB=> INSERT INTO students(username, email)
myDB-> VALUES ('student2', 'student2@mail.com');
INSERT 0 1
myDB=>
myDB=>
myDB=> SELECT * from students;
 user_id | username |      email
---------+----------+-------------------
       1 | student1 | student1@mail.com
       2 | student2 | student2@mail.com
(2 rows)
```

Network Access Control Lists (NACL)

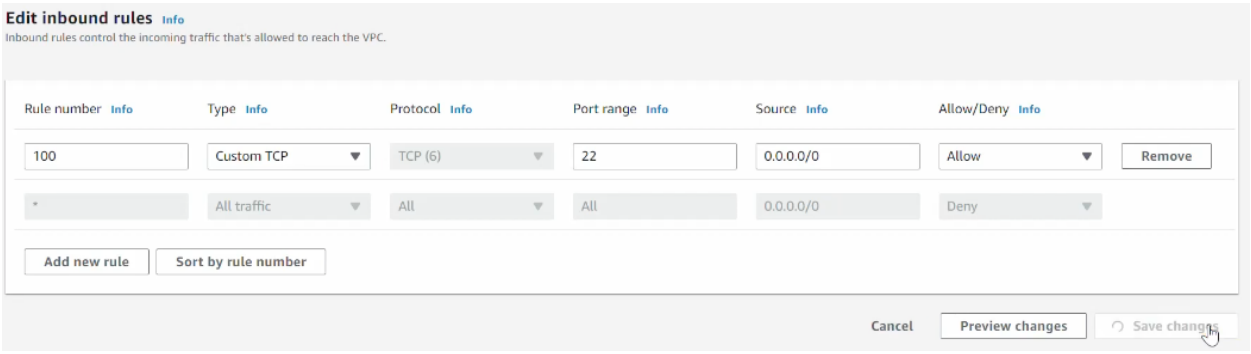
NACL sú sieťové prístupové zoznamy, ktoré je možné používať na obmedzenie komunikácie medzi podsieťami. NACL majú stateless správanie a teda pravidlo pridané do NACL automaticky nepovoľuje spätnú komunikáciu. Princíp je rovnaký ako pri ACL a teda sa jedná o číslovaný zoznam pravidiel, kde je možné mať **Allow** a **Deny** pravidlá. NACL sa vyhodnocuje zhora nadol, pričom na konci je implicitný záznam deny any. K ACL sa dá dostať cez záložku **VPC, Security** a **Network ACLs,** kde v rámci ACL je možné vidieť jeho základnú charakteristiku, vstupné a výstupné pravidla. Nakoniec je súčasťou ACL aj jeho asociácia s podsieťami, na ktoré bude NACL aplikovaný.

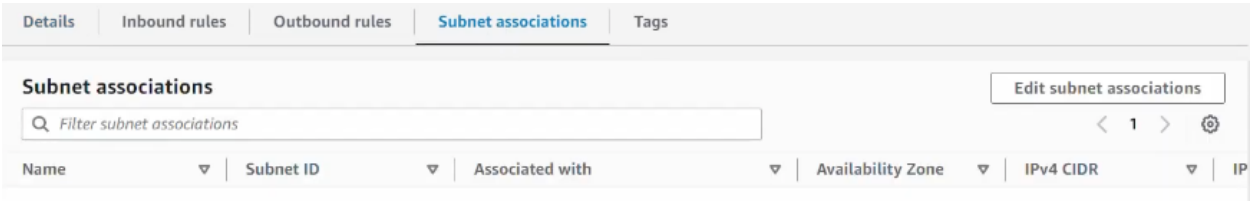


Pri vytváraní nového ACL stačí kliknúť na **Create network ACL** a nastaviť jeho názov a VPC do ktorej bude patriť.



V novo vytvorenom ACL je potrebné vložiť pravidlá podľa požiadaviek a následne stačí zvoliť z ponuky dostupných podsietí, do ktoreých bude NACL patriť. Povolenie SSH pripojenia z akéhokoľvek zdroja znázorňuje nasledujúci obrázok:

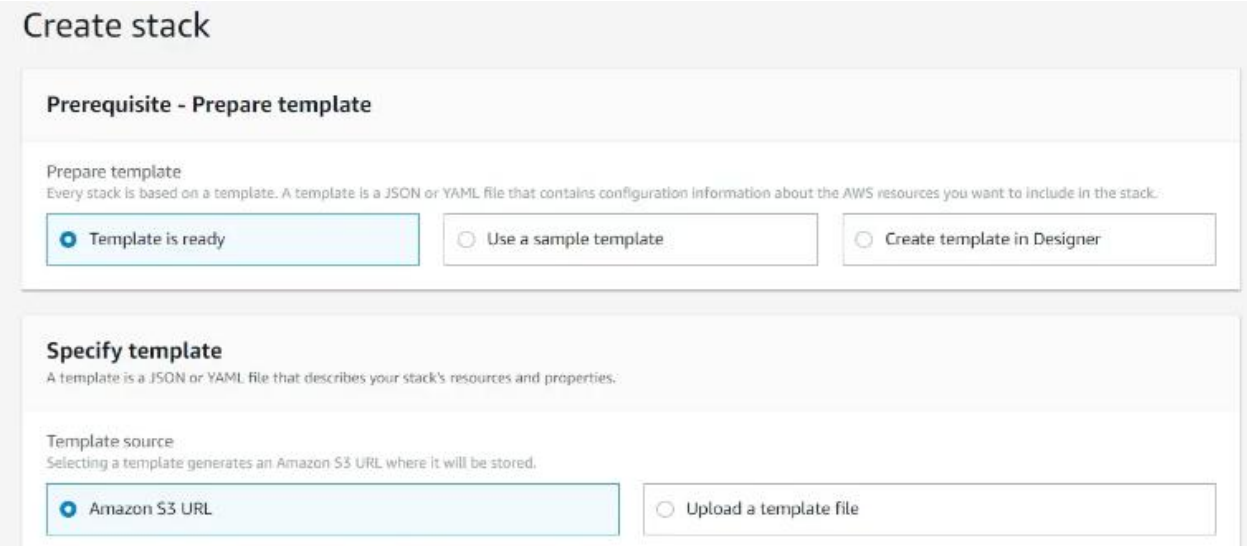




CloudFormation

CloudFormation je služba na automatizované nasadzovanie zdrojov založenom na definovaní konfiguračného súboru v YAML alebo JSON formáte. Tento konfiguračný súbor stačí následne

vložiť v rámci vytvárania **Stack** objektu v kontexte CloudFormation služby. Pri vytváraní Stack-u je možné zvoliť použitie šablóny a miesto, kde sa nachádza (S3 alebo na lokálnom PC).



Po vytvorení Stack objektu dôjde k vytvoreniu všetkých jeho zdrojov. Výhodou tohto spôsobu práce so zdrojmi je, že ich je možné priebežne aktualizovať a hlavne na konci, pri zmazaní celého Stack objektu dôjde k vymazaniu všetkých zdrojov, vďaka čomu nedôjde k zabudnutiu spustenie niektorej zo služieb, ktorej nechcená prevádzka môže spôsobiť značné finančné dôsledky. Nasledujúci výpis znázorňuje základný konfiguračný súbor pre vytvorenie vybraných vyššie uvedených zdrojov (VPC, SUB_A, SUB_B, EC2, SG, IGW):

```
Resources:
  MyVPC:
    Type: AWS::EC2::VPC
    Properties:
      CidrBlock: "10.0.0.0/16"
      EnableDnsSupport: 'true'
      EnableDnsHostnames: 'true'
      Tags:
       - Key: "Name"
         Value: "VPC"

  MySubnet1:
    Type: AWS::EC2::Subnet
    Properties:
      VpcId: !Ref MyVPC
      CidrBlock: "10.0.10.0/24"
      AvailabilityZone: us-east-1a
      Tags:
        - Key: "Name"
          Value: "SUB_A"
```

```
  MySubnet2:
    Type: AWS::EC2::Subnet
    Properties:
      VpcId: !Ref MyVPC
      CidrBlock: "10.0.20.0/24"
      AvailabilityZone: us-east-1b
      Tags:
        - Key: "Name"
          Value: "SUB_B"

  MySecurityGroup:
    Type: AWS::EC2::SecurityGroup
    Properties:
      VpcId: !Ref MyVPC
      GroupDescription: "Allow SSH and HTTP traffic"
      SecurityGroupIngress:
        - IpProtocol: tcp
          FromPort: 22
          ToPort: 22
          CidrIp: "0.0.0.0/0"
        - IpProtocol: tcp
          FromPort: 80
          ToPort: 80
          CidrIp: "0.0.0.0/0"
        - IpProtocol: icmp
          FromPort: -1
          ToPort: -1
          CidrIp: "0.0.0.0/0"
      Tags:
        - Key: "Name"
          Value: "SG"

  MyEC2Instance1:
    Type: AWS::EC2::Instance
    Properties:
      InstanceType: t2.micro
      ImageId: ami-04cb4ca688797756f  # Replace with a valid AMI ID
      KeyName: !Ref MyKeyPair
      NetworkInterfaces:
      - AssociatePublicIpAddress: "true"
        DeviceIndex: "0"
        GroupSet:
          - Ref: "MySecurityGroup"
        SubnetId:
          Ref: "MySubnet1"
      Tags:
        - Key: "Name"
          Value: "PC1"

  MyKeyPair:
    Type: "AWS::EC2::KeyPair"
```

```
    Properties:
      KeyName: "my-keypair"

  MyInternetGateway:
    Type: AWS::EC2::InternetGateway
    Properties:
      Tags:
        - Key: Name
          Value: MyInternetGatewayName

  AttachGateway:
    Type: AWS::EC2::VPCGatewayAttachment
    Properties:
      VpcId: !Ref MyVPC
      InternetGatewayId: !Ref MyInternetGateway
```