



# APACHE CASSANDRA DARBINĀŠANAI NEPIECIEŠAMĀS INFRASTRUKTŪRAS IZVEIDĒ AWS MĀKONĪ

D.Gabaliņa, R.Jankovskis

**DEVOPS PAMATI IESĀCĒJIEM**

**2022.gada 4.jūlijs**

**[https://github.com/DitaGabalina/  
devops\\_db\\_group](https://github.com/DitaGabalina/devops_db_group)**

# MĒRĶIS

Praktiski izmēģināt kursā apgūtos rīkus scenārijā, kāds tiešā veidā netika mācīts

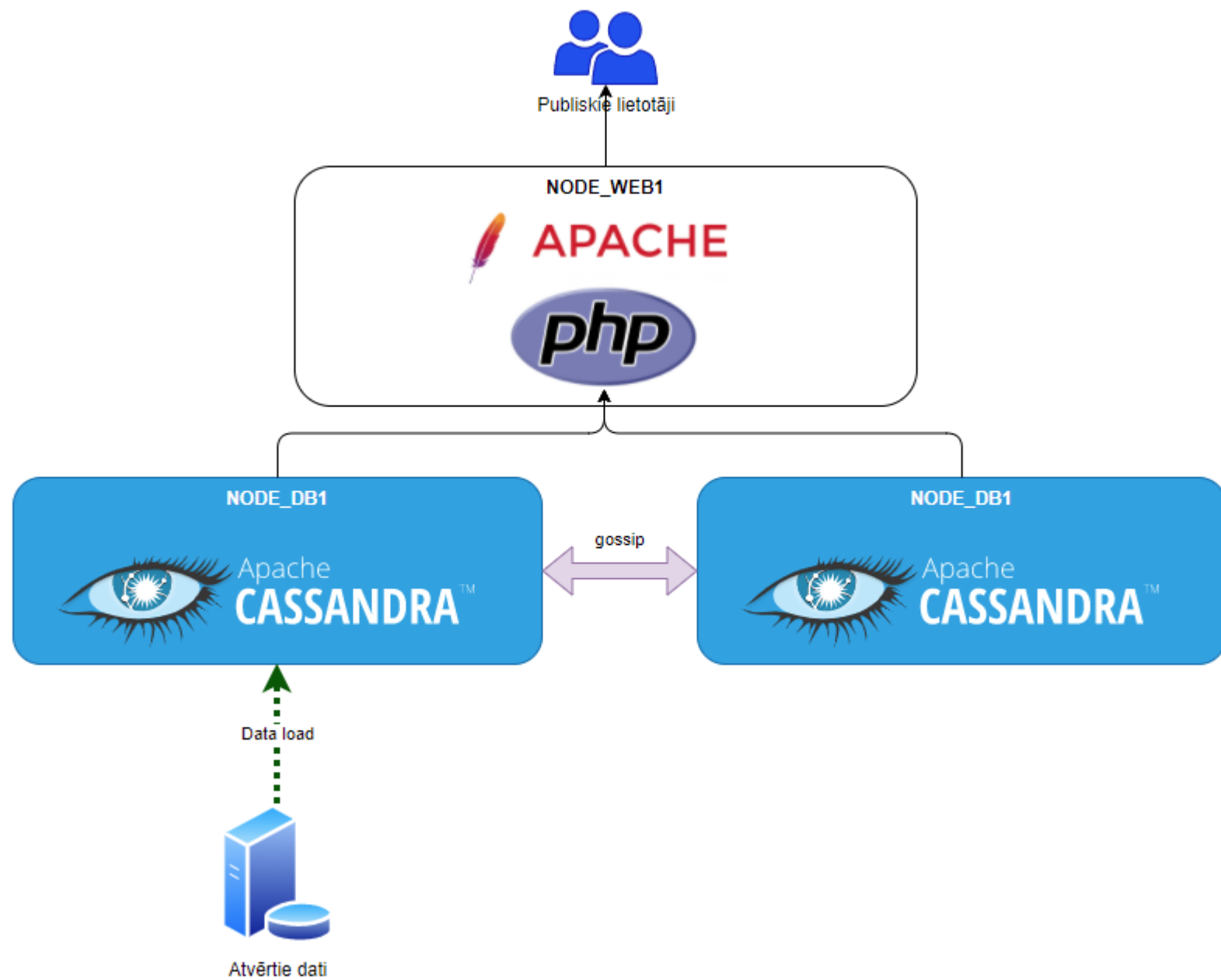
## Veidojamais risinājums

### ▢ Minimālā programma

- ▢ Izveidot infrastruktūras risinājumu, kas uzstāda standartprogrammatūru, piestartē datu bāzi un ielādē tajā datus no kāda atvērto datu avota

### ▢ Jauki ja būtu («could have» prioritāte)

- ▢ Apakštīkls, nodalot publisko piekļuvi un privātā tīklā – datu bāzes nodes
- ▢ Datu bāzes nodes apvienotas klāsterī un spēj izkļiedēt datus ar Gossip protokolu
- ▢ Iegūtie dati tiek atrādīti lietotājam, izmantojot web serveri
- ▢ Piesaistīt AWS publiskās IP adreses, lai tās nemainās pēc serveru pārstartēšanas vai infrastruktūras pārbūves
- ▢ Izmēģināt GitLab Runner / Github actions



# IZMANTOTIE RĪKI UN RISINĀJUMI

AWS – mākoņskaitļošanas pakalpojumu piegādātājs (EC2, EIP, VPC)

GIT – pirmkoda kontrole, projekta darbi

Terraform – infrastruktūras izveide

Ansible – programmatūras instalācija uz izveidotajiem serveriem

Ubuntu – uzstādāmā OS, Linux paveids, 22.04 versija (t3.small)

OpenSSH – atslēgu ģenerēšana SSH piekļuvei

Apache Cassandra – klāsterēts datu bāzu risinājums augstai pieejamībai, CQLSH – komandrindas rīks

Apache webserver (httpd) – lietojumserveris

CURL – datņu lejupielādes rīks

Dati.ur.gov.lv – lejupielādējamo datu avots

GitHub, GitHub wiki, GitHub projects

Telegram



# IZMANTOTIE RĪKI UN RISINĀJUMI

## Ditas izstrādes un testa vide

- ▢ Windows 10
- ▢ Visual Studio Code – teksta redaktors, darbs ar Git
- ▢ Tortoise Git – Git komandas no Windows Explorer
- ▢ Windows Terminal – SSH piekļuve izveidotajiem serveriem
- ▢ VMWARE, VM ar Oracle Linux 8 (Mēģinājums izveidot terraform un aws darbināšanai Linux vidē)
- ▢ Ditas AWS konts
- ▢ GitLab
- ▢ MkDocs (tehnisko piezīmju krāšanai)

## Rolanda izstrādes un testa vide

- Windows 11
- Ubuntu 22.04/18.04 LTS
- Visual Studio Code – terraform un ansible izstrāde ar tam paredzētajiem paplašinājumiem
- Visual Studio Code – remote ssh host, playbook izstrādei iekš AWS remote hostiem
- Windows Terminal
- Github Desktop
- Git bash / GUI
- AWS lu-vumc-devops konts

# PROJEKTĀ IZMANTOTĀS LABĀS PRAKSES UN SECINĀJUMI PIRMKODA PĀRVALDĪBA

Pirmkoda pārvaldība tika veikta Git, replicējot tās uz GitLab un GitHub attālinātajām repozitorijām

Pirmkoda pārvaldībai tika veidoti zari - pēc kolēģa veiktās koda pārskates *feature* zari tika sapludināti ar *development* zaru

GitHub tika izveidotas *wiki* lapas, kas apraksta projekta mērķi, veidojamo infrastruktūru, zarošanas stratēģiju

# PROJEKTĀ IZMANTOTĀS LABĀS PRAKSES UN SECINĀJUMI KOMANDAS DARBA VADĪBA

Darba noslodzes dēļ ne visi komandas biedri varēja pilnvērtīgi iesaistīties kopējos darbos, (ar šādu risku jāreķinās, un tas jāvada, uzstādot prasību prioritātes un mērķus, minimālo programmu)

Palīdzēja GitHub projects dashboard - kopējais darbu saraksts (paldies Pāvelam) , kuru izmantojām lai sadalītu darbus un atsekotu to statusu

Agile un Devops ir iteratīvi procesi, tie vienmēr uzlabojas, un arī tika uzlaboti savstarpējā saziņā

Vēlams savlaicīgi vienoties par *naming conventions*, jo pēcāk tos būs grūti labot

# PROJEKTĀ IZMANTOTĀS LABĀS PRAKSES UN SECINĀJUMI INFRASTRUKTŪRA KĀ KODS (IOC)

Izmantotie rīki (Terraform un Ansible) ļauj elastīgi veidot infrastruktūru un šim nolūkam nepieciešamo informāciju glabāt pirmkoda pārvaldības sistēmās

Komandas, kas tiek izmantotas serveru instalēšanai, vēlams pirms tam pārbaudīt un tikai tad likt IOC skriptos

Svarīgi pievērst uzmanību Linux piekļuves tiesībām (ar kādu lietotāju tiek veikta instalācija, kādas tam ir piekļuves tiesības, kam jāizmanto sudo, kādas komandas ar sudo var veikt bez atkārtotas paroles ievadīšanas, failu un mapju *permissions*)

SSH atslēgas (nesaputroties, tās var būt vairākas, netīšām neiekļaut pirmkodā)



# CASSANDRA | DATU LEJUPIELĀDE

4 lines (4 slo) | 531 Bytes

Raw

Blame



```
1 curl https://dati.ur.gov.lv/register/register.csv --output ~/external_data/register.csv --silent --insecure
2 curl https://dati.ur.gov.lv/register/ppi_public_persons_institutions.csv --output ~/external_data/ppi_public_persons_institutions.csv --silent --insecure
3 curl https://dati.ur.gov.lv/register/register_name_history.csv --output ~/external_data/register_name_history.csv --silent --insecure
4 curl https://dati.ur.gov.lv/register/ppi_delegated_entities.csv --output ~/external_data/ppi_delegated_entities.csv --silent --insecure
```

# CASSANDRA | DATU MODELIS

76 lines (70 sloc) | 1.89 KB

Raw

Blame



```
1  -- Create a keyspace - replication factor should not exceed the number of nodes in your cluster
2  CREATE KEYSPACE IF NOT EXISTS ur WITH REPLICATION = { 'class' : 'SimpleStrategy', 'replication_factor' : '2' };
3
4  -- data_load\register_name_history.csv
5  CREATE TABLE IF NOT EXISTS ur.register_name_history (
6    regcode text PRIMARY KEY,
7    name text,
8    date_to timestamp
9  );
10
11 -- data_load\ppi_delegated_entities.csv
12 CREATE TABLE IF NOT EXISTS ur.ppi_delegated_entities (
13   registrationNumber text PRIMARY KEY,
14   name text,
15   delegatedEntityName text,
16   delegatedEntityRegistrationNumber text,
17   delegatedEntityRegisteredOn timestamp,
18   delegatedEntityRemovedOn timestamp
19 );
20
```

# CASSANDRA | DATU IELĀDE

4 lines (3 sloc) | 212 Bytes

Raw

Blame



```
1
2 TRUNCATE ur.register_name_history ;
3 -- Insert some data
4 COPY ur.register_name_history (regcode,date_to,name) FROM './register_name_history.csv' WITH HEADER = TRUE AND DELIMITER=';' and DATETIMEFORMAT='%Y-%m-%d';
```

# CASSANDRA | REZULTĀTA PĀRBAUDE

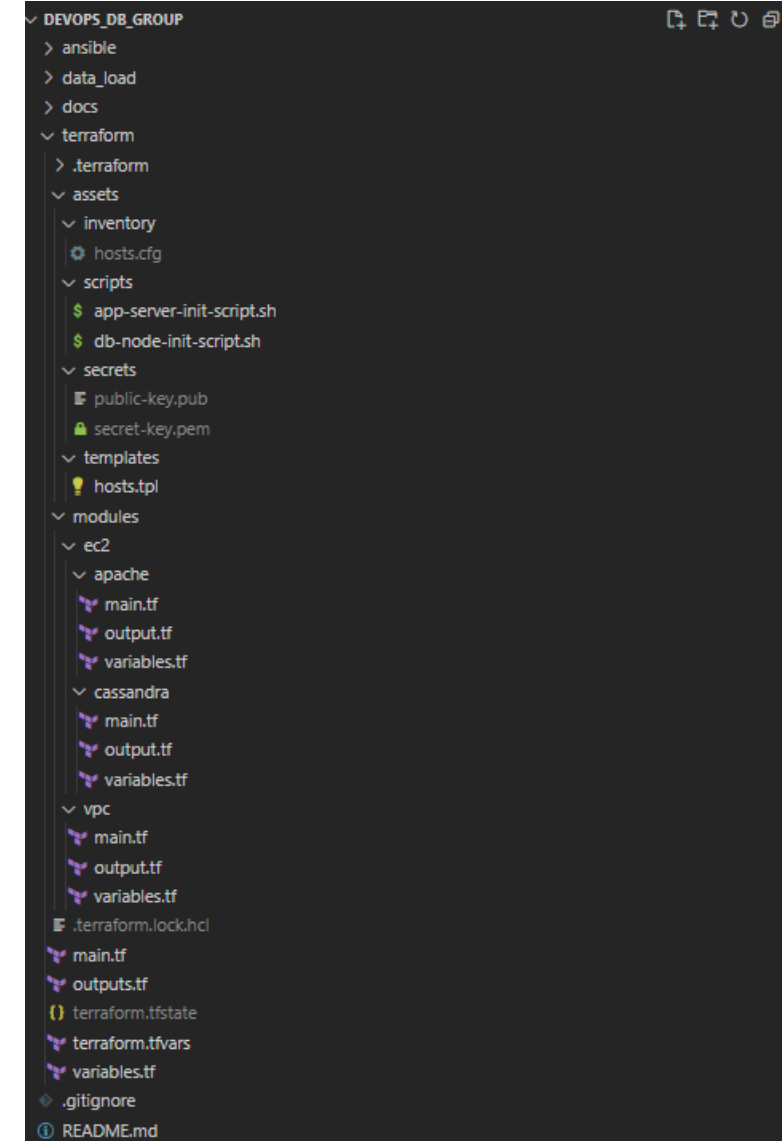
```
select * from ur.register_name_history;
```

40002086696	2003-09-17 00:00:00.000000+0000	SIA "TRIALS" ""
42103048376	2010-07-14 00:00:00.000000+0000	SIA "L Lingerie" ""
45403022053	2017-01-12 00:00:00.000000+0000	Sabiedrība ar ierobežotu atbildību "MP Skaistums" ""
40002079092	2004-02-16 00:00:00.000000+0000	Ogres rajona A.Grašiņas individuālais uzņēmums "BIRZGALES APTIEKA" ""
43603041468	2021-08-04 00:00:00.000000+0000	Sabiedrība ar ierobežotu atbildību "Andas Kaires ārsta prakse zobārstniecībā" ""
40008018867	2004-12-13 00:00:00.000000+0000	Aizgādņu fonds "ASSOR" ""
40103669322	2015-10-22 00:00:00.000000+0000	SIA "4 b.e.e." ""
40003051329	2004-03-03 00:00:00.000000+0000	Sabiedrība ar ierobežotu atbildību-kopuzņēmums firma "EKJU" ""
40002036661	2004-12-08 00:00:00.000000+0000	Valērija Černišova individuālais uzņēmums firma "BESTS" ""
44101034647	2006-07-03 00:00:00.000000+0000	Mārsnēnu pagasta Ulda Pilvera zemnieka saimniecība "UPMAĻI" ""
40003114200	2003-04-02 00:00:00.000000+0000	Ražošanas komercfirma sabiedrība ar ierobežotu atbildību "MD VĀRME" ""
40003087504	2004-12-28 00:00:00.000000+0000	Sabiedrība ar ierobežotu atbildību "KRAUJA" ""
40003239883	1995-01-09 00:00:00.000000+0000	Sabiedrība ar ierobežotu atbildību "AGROBIZNES" ""
40003898810	2008-05-06 00:00:00.000000+0000	Sabiedrība ar ierobežotu atbildību "MADONAS PROJEKTS" ""
40003397454	2001-10-04 00:00:00.000000+0000	Sabiedrība ar ierobežotu atbildību "Š UN C" ""
46601010028	2005-10-05 00:00:00.000000+0000	Limbažu rajona Ainažu pilsētas zemnieku saimniecība "AVOTKALNI" ""
40003791008	2012-01-13 00:00:00.000000+0000	Sabiedrība ar ierobežotu atbildību "KD Jet" ""
40008044018	2008-10-10 00:00:00.000000+0000	Latvijas Aviācijas Darbinieku Arodbiedrību Federācija
50003263641	1995-10-25 00:00:00.000000+0000	INŽENIERTEHNISKĀ FIRMA "ALKORS" ""
40003214019	2004-08-09 00:00:00.000000+0000	SIA kopuzņēmums "UNITED OILS" ""
44102013040	2009-01-12 00:00:00.000000+0000	V.Zikovas individuālais uzņēmums

# Terraform struktūra

Terraform direktorija satur:

- root direktorijā terraform failus, kur definēti un sasaistīti visi terraform moduļi, zem /modules direktorijas
- *modules* direktorija ir visi nepieciešamie terraform moduļi:
  - EC2 direktorija satur 2 moduļus - *apache* un *cassandra*
  - VPC modulis satur visu nepieciešamo tīkla izveidei iekš *aws* *apache* un *cassandra* moduļiem.
- *assets* direktorijā tiek glabāts viss pārējais, kas nepieciešams terraform, piemēram template, ssh atslēgas un skripti



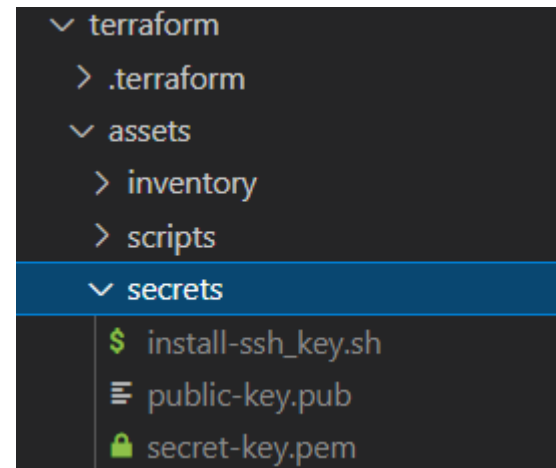
# Terraform palaišana

Lai palaistu Terraform nepieciešams atvērt root direktorijā esošo terraform.tfvars un nomainīt

- aws-private-key-location uz ceļu kur atrodās jūsu aws privātā atslēgu un
- aws-access-key-name uz jūsu aws ģenerētā key-pair nosaukumu
- Papildus arī nepieciešams ģenerēt ssh atslēgas ansible master (privāto) un ansible host (publisko), kā arī ssh ielādēšanas skriptu
- Esot projekta root direktorijā to var izdarīt ar powershell komandām:

```
mkdir .\terraform\assets\secrets | cd
ssh-keygen -t rsa -N -q -f secret-key.pem -N ''
ren .\secret-key.pem.pub public-key.pub
echo "#!/bin/bash" >> install-ssh_key.sh ; echo "echo '$(cat public-key.pub)' >> /home/ubuntu/.ssh/authorized_keys" >> install-ssh_key.sh
```

```
terraform > terraform.tfvars > ...
1   region="us-west-1"
2   vpc-cidr="192.168.0.0/16"
3   env="Cassandra_test_env"
4   public_subnets_cidr = ["192.168.1.0/24", "192.168.2.0/24"]
5   private_subnets_cidr = ["192.168.3.0/24", "192.168.4.0/24"]
6   availability_zones   = ["us-west-1a", "us-west-1c"]
7   instance_type        = "t3.small"
8   aws-private-key-location = "~/rolands_jankovskis.pem"
9   aws-access-key-name    = "rolands_jankovskis"
```



# Ansible uzstādīšana caur Terraform

## (ansible hosts ģenerēšana)

```
resource "local_file" "hosts_cfg" {  
  content = templatefile("./assets/templates/hosts.tpl",  
    {  
      cassandra_node1 = aws_instance.cassandra_db_node_1.private_ip  
      cassandra_node2 = aws_instance.cassandra_db_node_2.private_ip  
    }  
  )  
  filename = "./assets/inventory/hosts.cfg"  
}
```

```
hosts.tpl x  
terraform > assets > templates > hosts.tpl  
1 [cassandra_nodes]  
2 ${cassandra_node1}  
3 ${cassandra_node2}  
4  
5 [cassandra_main_node]  
6 ${cassandra_node1}  
7  
8 [cassandra_secondary_node]  
9 ${cassandra_node2}  
10
```

- Iekš Terraform moduļa definējam lokālā faila resursu kurā tiek norādīts lai izvada privātās ip adreses
- Privātās ip adreses tiek padotas uz hosts.tpl un pēc tā satura tiek izveidots fails – hosts.cfg

\* Iespējams arī ģenerēt *template*, lai tiek aizsūtītas visas moduļu instanču adreses izmantojot “for loop”, automatizētai *inventory* izveidei

```
hosts.cfg x  
terraform > assets > inventory > hosts.cfg  
1 [cassandra_nodes]  
2 192.168.3.28  
3 192.168.3.169  
4  
5 [cassandra_main_node]  
6 192.168.3.28  
7  
8 [cassandra_secondary_node]  
9 192.168.3.169  
10
```

# Ansible uzstādīšana caur Terraform

## (ansible master konfigurēšana)

1

Provisioner File caur ssh nosūta privāto atslēgu uz ansible master, node pārvaldīšanai un ģenerēto ansible inventory — hosts.cfg

```
provisioner "file" {
  source = "./assets/secrets/secret-key.pem"
  destination = "/home/ubuntu/.ssh/id_rsa"

  connection {
    type = "ssh"
    user = "ubuntu"
    private_key = file("${var.aws_private_key}")
    host = self.public_dns
  }
}

provisioner "file" {
  source = "./assets/inventory/hosts.cfg"
  destination = "/tmp/hosts"

  connection {
    type = "ssh"
    user = "ubuntu"
    private_key = file("${var.aws_private_key}")
    host = self.public_dns
  }
}
```

2

- Uz Ansible Master tiek uzstādītas pareizās tiesības privātajai ssh atslēgai, uzstādīts ansible, pareizi instalēts sagavaotais inventory fails.
- Tad tiek klonēts repozitorijs un startēti sagatavotie playbooki

```
}
provisioner "remote-exec" {
  inline = [
    "sudo chmod 400 ~/.ssh/id_rsa",
    "sudo apt-add-repository ppa:ansible/ansible -y",
    "sudo apt update -y",
    "sudo apt install ansible -y",
    "sudo mv /etc/ansible/hosts /etc/ansible/hosts.bak",
    "sudo mv /tmp/hosts /etc/ansible/hosts",
    "mkdir ~/ansible-codes",
    "git clone -b development https://github.com/DitaGabalina/devops_db_group.git ~/ansible-codes/devops_db_group",
    "ansible-playbook ~/ansible-codes/devops_db_group/ansible/cassandra_node-setup.yaml --ssh-common-args='-o StrictHostKeyChecking=accept-new'",
    "ansible-playbook ~/ansible-codes/devops_db_group/ansible/cassandra_main_node-data_load.yaml",
    "ansible-playbook ~/ansible-codes/devops_db_group/ansible/cassandra-cluster-creation.yaml",
    "ansible-playbook ~/ansible-codes/devops_db_group/ansible/apache_server-setup.yaml --ssh-common-args='-o StrictHostKeyChecking=accept-new'"
  ]
}

connection {
  type = "ssh"
  user = "ubuntu"
  private_key = file("${var.aws_private_key}")
  host = self.public_dns
}
```



# Ansible uzstādīšana caur Terraform

## (ansible master konfigurēšana)

1

Provisioner File caur ssh nosūta privāto atslēgu uz ansible master, node pārvaldīšanai un ģenerēto ansible inventory — hosts.cfg

```
provisioner "file" {
  source = "./assets/secrets/secret-key.pem"
  destination = "/home/ubuntu/.ssh/id_rsa"

  connection {
    type = "ssh"
    user = "ubuntu"
    private_key = file("${var.aws_private_key}")
    host = self.public_dns
  }
}

provisioner "file" {
  source = "./assets/inventory/hosts.cfg"
  destination = "/tmp/hosts"

  connection {
    type = "ssh"
    user = "ubuntu"
    private_key = file("${var.aws_private_key}")
    host = self.public_dns
  }
}
```

2

- Uz Ansible Master tiek uzstādītas pareizās tiesības privātajai ssh atslēgai, uzstādīts ansible, pareizi instalēts sagavaotais inventory fails.
- Tad tiek klonēts repozitorijs un startēti sagatavotie playbooki

```
}
provisioner "remote-exec" {
  inline = [
    "sudo chmod 400 ~/.ssh/id_rsa",
    "sudo apt-add-repository ppa:ansible/ansible -y",
    "sudo apt update -y",
    "sudo apt install ansible -y",
    "sudo mv /etc/ansible/hosts /etc/ansible/hosts.bak",
    "sudo mv /tmp/hosts /etc/ansible/hosts",
    "mkdir ~/ansible-codes",
    "git clone -b development https://github.com/DitaGabalina/devops_db_group.git ~/ansible-codes/devops_db_group",
    "ansible-playbook ~/ansible-codes/devops_db_group/ansible/cassandra_node-setup.yaml --ssh-common-args='-o StrictHostKeyChecking=accept-new'",
    "ansible-playbook ~/ansible-codes/devops_db_group/ansible/cassandra_main_node-data_load.yaml",
    "ansible-playbook ~/ansible-codes/devops_db_group/ansible/cassandra-cluster-creation.yaml",
    "ansible-playbook ~/ansible-codes/devops_db_group/ansible/apache_server-setup.yaml --ssh-common-args='-o StrictHostKeyChecking=accept-new'"
  ]
}

connection {
  type = "ssh"
  user = "ubuntu"
  private_key = file("${var.aws_private_key}")
  host = self.public_dns
}
```

# Ansible uzstādīšana caur Terraform

## (ansible host konfigurēšana)

- Izmantojot user-data komandu tiek palaists sagatavots skripts kas uzkopē publisko atslēgu pie authorized keys.

```
resource "aws_instance" "cassandra_db_node_2" {
  ami           = data.aws_ami.ubuntu.id
  instance_type = var.instance_type
  key_name      = "${var.aws-keypair-name}"
  subnet_id    = var.subnet_id[0]
  vpc_security_group_ids = var.vpc_security_group_idsc
  tags = {
    Name = "cassandra_db_node_2"
  }
  user_data = "${file("./assets/secrets/install-ssh_key.sh")}"
}
```

\* Šajā scenārijā remote-exec un file provideris nav pieļaujams, kā ansible master, jo instances atrodās privātā tīklā un tām pa tiešo nevar pieslēgties

# Instanču uzstādīšana un konfigurēšana caur Ansible

(ansible master Apache2 un php uzstādīšana)

- Norādīts hosts – localhost, lai izpildītos uz ansible master
- Uzstāda apache2 un php uz master node

```
---  
- hosts: localhost  
  become: true  
  tasks:  
    - name: Update and upgrade apt packages  
      apt:  
        upgrade: yes  
        update_cache: yes  
    - name: Install Apache  
      package:  
        name: apache2  
        state: latest  
    - name: Start theApache service  
      service:  
        name: apache2  
        state: started  
    - name: Install Php  
      package:  
        name: php  
        state: present
```

# Instanču uzstādīšana un konfigurēšana caur Ansible

(ansible node Apache Cassandra uzstādīšana ar dependencies un java test)

- uzstāda nepieciešamās apt pakas
- instalē php atslēgas priekš Cassandra repozitorija
- uzstāda Cassandra
- iestata env mainīgo JAVA\_HOME un to pārbauda

```
---
- hosts:
  - cassandra_nodes
  become: true
  tasks:
    - name: Update and upgrade apt packages
      apt:
        upgrade: yes
        update_cache: yes
    - name: Install MC
      package:
        name: mc
        state: latest

    - name: Install openjdk-11-jre-headless
      package:
        name: openjdk-11-jre-headless
        state: present

    - name: Install apt-transport-https
      package:
        name: apt-transport-https
        state: present

    - name: Add Key for cassandra repo
      apt_key:
        url: https://www.apache.org/dist/cassandra/KEYS
        state: present

    - name: Add cassandra debian repository
```

```
- name: Add cassandra debian repository
  apt_repository: repo='deb http://www.apache.org/dist/cassandra/debian 40x main'

- name: Install cassandra
  package:
    name: cassandra
    state: present

- name: make sure cassandra is started
  service: name=cassandra state=restarted

- name: Set JAVA_HOME
  lineinfile:
    dest: /etc/environment
    state: present
    regexp: '^JAVA_HOME'
    line: 'JAVA_HOME=/usr/lib/jvm/java-11-openjdk-amd64'

- name: JAVA_HOME ielasisana
  shell: printf $JAVA_HOME
  register: output

- name: JAVA_HOME Parbaude
  debug:
    msg: "JAVA_HOME Veiksmīgi registrets"
  when: output.stdout is match ("/usr/lib/jvm/java-11-openjdk-amd64")
```

# Instanču uzstādīšana un konfigurēšana caur Ansible

(ansible datu lejupielāde un imports iekš main node)

- Izveido directory datu saglabāšanai
- Lejupielādē Github repo uz cassandra nodes
- No lejupielādētajiem datiem palaiž sagatavotos skriptus datu lejupielādei un importēšanai

```
---
- hosts:
  - cassandra_main_node
  tasks:
    - name: Create External_data dir
      file:
        path: ~/external_data
        state: directory
    - name: Download devops_db_group from the GitRepo
      git:
        repo: 'https://github.com/DitaGabalina/devops_db_group.git'
        dest: "~/devops_db_group/"
        version: development
        force: yes

    - name: Download UR Data
      shell: chmod +x ~/devops_db_group/data_load/download_ur_data.sh && ~/devops_db_group/data_load/download_ur_data.sh

    - name: Create Data Model
      shell: cqlsh -f ~/devops_db_group/data_load/create_cassandra_data_model.cql
      args:
        chdir: ~/external_data

    - name: Copy to cassandra
      shell: cqlsh -f ~/devops_db_group/data_load/copy_to_cassandra.cql
      ignore_errors: yes
      args:
        chdir: ~/external_data
```

# Instanču uzstādīšana un konfigurēšana caur Ansible

## (ansible Cassandra node cluster izveide)

- Ielasa galvenās nodes ip adresi uz mainīgo seeds
- Aizstāj konfigurācijas failu cassandra.yaml uz ansible hostiem pielietojot sagatavoto šablonu un mainīgos
- Restartē cassandru, pagaida kamēr tā iestartējas un tad palaiž repair kommandu lai dati tiktu replicēti.

```
---
- hosts:
  - cassandra_nodes

  tasks:
    - name: Create List of Ip addresses to be added to cassandra.yaml
      set_fact: seeds="{{ groups['cassandra_main_node'] | map('extract', hostvars, ['ansible_default_ipv4', 'address']) | join(',') }}"

    - name: override cassandra.yaml file with one created from template
      become: true
      template: src=./assets/templates/cassandra.yaml dest=/etc/cassandra/

    - name: make sure cassandra is restarted
      become: true
      service: name=cassandra state=restarted

    - name: Pause for 1 minutes to wait for Cassandra service
      pause:
        minutes: 1

- hosts:
  - cassandra_main_node

  tasks:
    - name: Launch nodetool repair
      shell: nodetool repair
```

```
# Setting listen_address to 0.0.0.0 is always wrong.
#
listen_address: {{ inventory_hostname }}
```

```
seed_provider:
  # Addresses of hosts that are deemed contact points.
  # Cassandra nodes use this list of hosts to find each other and learn
  # the topology of the ring.  You must change this if you are running
  # multiple nodes!
  - class_name: org.apache.cassandra.locator.SimpleSeedProvider
    parameters:
      # seeds is actually a comma-delimited list of addresses.
      # Ex: "<ip1>,<ip2>,<ip3>"
      - seeds: "{{ seeds }}"
```

# Rezultāts pēc Terraform apply

- Pilns Terraform apply aizņem aptuveni 11 min un 40 sek, no kurām 5 min ir datu lejupielāde no UR
- Terraform automatizēti uzstāda ansible struktūru un apmaina ssh keys starp hostiem
- Terraform uzstāda VPC tīklu ar 2 privātiem un 2 publiskiem subnetiem kas ir pārdalīti pa 2 dažādiem availability zones
- Terraform/ansible rezultātā tiek uzstādītas 3 instances – 2 Cassandra nodes privātā tīkla un 1 Apache node publiskā tīklā
- Ansible uzstāda Apache server, Cassandra un tās dependencies un ielādē datus no atvērtiem datiem
- Ansible apvieno abas Cassandra nodes vienā klasteri ar datu replikāciju

The screenshot displays the results of a Terraform apply operation, showing database queries, instance status, and the Apache2 default page.

**Database Queries:**

```
ubuntu@ip-192-168-3-169:~$ cqlsh -e "select * from ur.regis
ter_name_history;" | more

regcode | date_to | name
-----|-----|-----
45401011303 | 2019-09-20 00:00:00.000000+0000 | Praulienas pagasta Nebar
es zemnieku saimniecība "AIZSILNIEKI"
42401005668 | 2000-11-28 00:00:00.000000+0000 | Rēzeknes rajona Ilzeskalna pag
asta zemnieku saimniecība "KALNLEJAS"
50008034331 | 2005-04-18 00:00:00.000000+0000 |
Latvijas dzimstības veicināšanas fonds
40003530590 | 2009-01-05 00:00:00.000000+0000 | Sabiedrība a
r ierobežotu atbildību "Koka Dizains"
40008262886 | 2019-05-02 00:00:00.000000+0000 |
Latvijas Alus Jūdzes Federācija
46601006018 | 1997-03-17 00:00:00.000000+0000 | Vidrižu paga
sta zemnieku saimniecība "JAUNALKŠŅI"
44103012184 | 1997-04-07 00:00:00.000000+0000 | Sab
iedrība ar ierobežotu atbildību "VIA"
50002063591 | 2001-07-25 00:00:00.000000+0000 | Rīgas pilsētas Samsūras individu
40003530590 | 2009-01-05 00:00:00.000000+0000 | Rīgas pilsētas Samsūras individu
```

**Instance Status:**

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IPv4
cassandra_db_...	i-07b318003d5d87bd7	Running	t3.small	2/2 checks passed	User: arnaws:i	us-west-1a	-	-
cassandra_db_...	i-05cd9c98b5165abb0	Running	t3.small	2/2 checks passed	User: arnaws:i	us-west-1a	-	-
apache_server	i-00824ea81084e501d	Running	t3.small	2/2 checks passed	User: arnaws:i	us-west-1a	ec2-54-177-210-236.us...	54.177.210

**Ansible Status:**

```
ubuntu@ip-192-168-1-204:~$ ansible all -m ping
192.168.3.28 | SUCCESS => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/bin/
  },
  "changed": false,
  "ping": "pong"
}
192.168.3.169 | SUCCESS => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/bin/
  },
  "changed": false,
  "ping": "pong"
}
```

**Apache2 Default Page:**

Apache2 Default Page

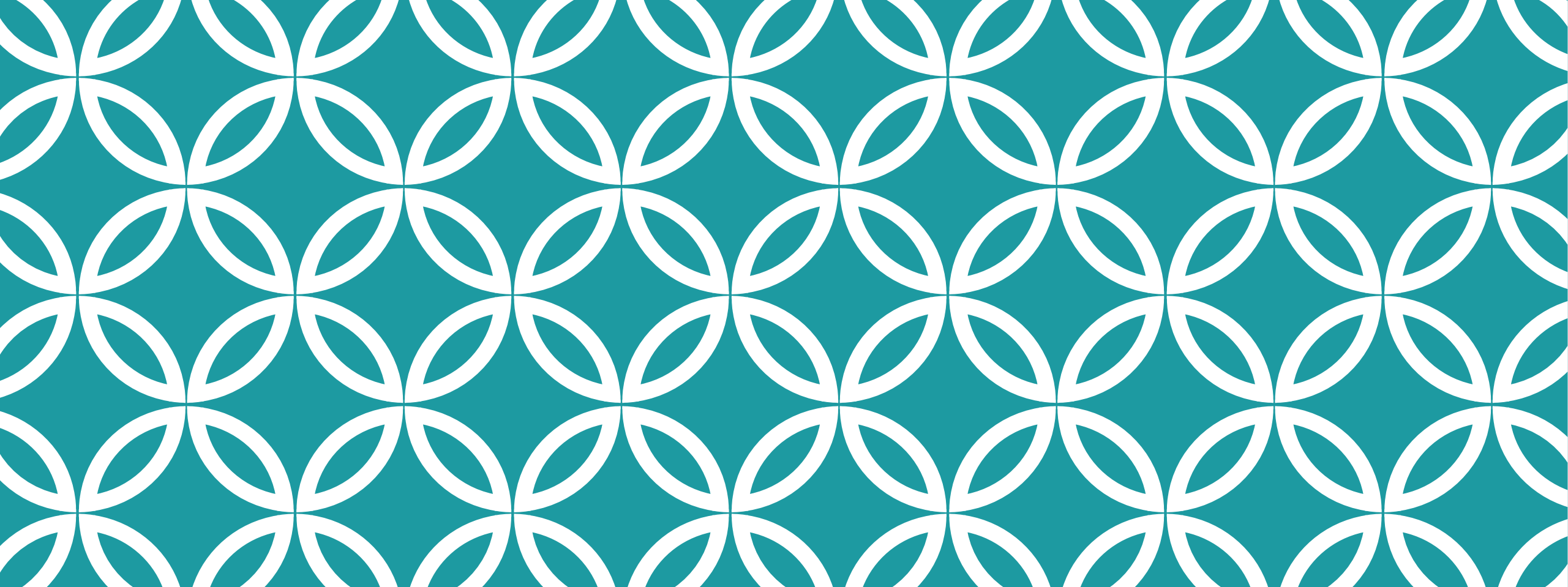
Ubuntu

It works!

This is the default welcome page used to test the correct operation of the Apache2 server after installation on Ubuntu systems. It is based on the equivalent page on Debian, from which the Ubuntu Apache packaging is derived. If you can read this page, it means that the Apache HTTP server installed at this site is working properly. You should **replace this file** (located at `/var/www/html/index.html`) before continuing to operate your HTTP server.

If you are a normal user of this web site and don't know what this page is about, this probably means that the site is currently unavailable due to maintenance. If the problem persists, please contact the site's administrator.

Configuration Overview



**PALDIES PAR  
UZMANĪBU!** |