# Mājasdarbs 05

2. grupa Māris Andersons

# Uzdevumi / manas atbildes / pieraksti iekrāsoti pelēkā krāsā

1. Atveriet savā pārlūkā <https://952122846739.signin.aws.amazon.com/console>vai <https://lu-vumc-devops.signin.aws.amazon.com/console>un ievadiet savus akreditācijas datus attiecīgajos laukos, t.i., lietotājvardu iekš “IAM user name” un paroli iekš “Password”.

ok

1. Atveriet “EC2” resursu, meklētājā uzrakstot “EC2” un uzspiežot uz tā nosaukumu.

ok

1. Augšējā labajā AWS konsoles stūrī uzspiediet uz reģiona nosaukumu un nomainiet to uz Jūsu grupai piemēroto reģionu:
   * 1. 1. grupa – us-east-1 (N. Virginia)
     2. 2. grupa – us-east-2 (Ohio)
     3. 3. grupa – us-west-1 (N. California)
     4. 4. grupa – us-west-2 (Oregon)

Graphical user interface, application

Description automatically generated

1. Kad esat izvēlējies pareizo reģionu – spiediet uz “Launch instances”.

ok

1. Izveidojiet trīs EC2 serverus manuālā veidā vai ar **Terraform**, izmantojot šādus parametrus:

a. “Name” jeb servera nosaukumā norādīt savu grupu un vārdu, piem.,

* + - * **2\_grupa\_ansible\_master\_maris\_andersons**
      * **2\_grupa\_ansible\_host\_1\_ maris\_andersons**
      * **2\_grupa\_ansible\_host\_2\_ maris\_andersons**
    1. “Instance type” izvēlēties: **t2.micro**
    2. OS tipu izvēlēties Ubuntu
    3. “Key pair (login)” izvēlēties savu key pair. 2\_grupa\_maris\_andersons-Ansible
    4. Pārējos iestatījumus atstāt pēc noklusējuma jeb nemainīt tos.
    5. Ok / izveidoju manuāli un izmantoju esošo key-pair, lai pa tianos piekļūtu savam repozitorijam Gitā

1. Nomainīt **hostname** katram serverim, attiecīgi:

sudo hostname master-maris-andersons

sudo hostname host-1-maris-andersons

sudo hostname host-2-maris-andersons

1. Uzstādiet **Ansible** programatūru uz master-*valdemars-kalejs*.

sudo apt update --yes

sudo apt install software-properties-common --yes

sudo add-apt-repository --yes --update ppa:ansible/ansible

sudo apt install ansible --yes

sudo apt upgrade --yes

papildus instalēju Ansible arg-complete package  
sudo apt install python3-argcomplete --yes

sudo activate-global-python-argcomplete3

ansible instalacijas parbaude -> ping komanda, kura pagaidām var atgriež tikai localhost

ansible all -m ping --ask-pass

1. Rediģēt **Ansible inventory** host pievienojot host instances. Izveidojot sekojošas serveru grupas.

**[servers]**

*<abi host serveri>*

**[webservers]**

*<hosts 1 >*

**[database]**

## <hosts 2>

Mkdir ansible-majasdarbs05

cp /etc/ansible/hosts ./ansible-majasdarbs05

cd ./ansible-majasdarbs05

ls

nano ./hosts un manuāli redigeju hosts failu, saglabājot oriģinālos komentārus

cat ./hosts

Text

Description automatically generated

1. Konfigurēt ssh atslēgu starp Master un host serveriem.

Šajā gadījumā ir iespēja izmantot AWS EC2 key-pair un nevajag taisīt pilnu key-pair izveides procesu. Uz instancēm jau ir veikta key-pair konfigurācija, es pārkopēju tikai privāto-atslēga uz Ansible-master datoru un salaboju tiesības.

touch ../.ssh/id\_rsa

nano ../.ssh/id\_rsa un manuāli iekopēju privāto atslēgu….

chmod -R go= ../.ssh

1. Pārbaudiet Ansible serveru konfigurāciju ar Ansible komandu ansible-inventory --list -y

ansible-inventory --list -y -i ./hosts

Text

Description automatically generated

1. Veiciet visu Ansible **host ping** ar Ansible ping komandu.

ansible all -m ping -i ./hosts

Text

Description automatically generated

1. Pārliecinieties ka ir attiecīgi konfigurēts **ansible.cfg**

Nesaprotu kas šeit jādara ?

1. Master node izveidojiet mapi kur glabāsiet savus Ansible kodu.

Mape tika izveidota pie 8.punkta un tajā glabājās hosts-fails

1. Izpildiet Ansible Ad hoc komandu lai nolasītu host serveros esošos procesus. Hint: Linux sistēmas procesi.

ansible -i ./hosts servers -a "sudo ps ax"

1. Noklonējiet repozitoriju uz Master nodes <https://github.com/lu-vumc-devops/lu-vumc-devops.ansible.git>

sudo apt install git –yes

cd /home/ubuntu/ansible-majasdarbs05/

git init

git clone https://github.com/lu-vumc-devops/lu-vumc-devops.ansible.git

1. Pārveidojiet failu AdditionalPackages.yml ta lai http tiktu uzstādīts uz host1 un datubāzes uz host2. Veiciet testu. Hint: playbook failu var sadalīt vairākos.

nano lu-vumc-devops.ansible/AdditionalPackages.yml // rediģēju failu

ansible-playbook -i hosts --syntax-check ./lu-vumc-devops.ansible/AdditionalPackages.yml // sintakses pārbaude atrada kļūdas un vajadzēja laiku to izlabot

ansible-playbook -i hosts -C ./lu-vumc-devops.ansible/AdditionalPackages.yml // dry-run atrada kļūdu, ka pietrūkst sudo-tiesības

// apt-get pieprasīja, lai “httpd package” tiktu norādīts precīzs nosaukums

ansible-playbook -i hosts ./lu-vumc-devops.ansible/AdditionalPackages.yml // instalācija notika veiksmīgi

Text

Description automatically generated

Text

Description automatically generated

1. Attiecīgi pārveidojiet AdditionalPackagesDelete.yml. lai delete notiktu tajos serveros kur iepriekš tika uzstādīts http un mariadb. Veiciet testu.

nano lu-vumc-devops.ansible/AdditionalPackagesDelete.yml

ansible-playbook -i hosts ./lu-vumc-devops.ansible/AdditionalPackagesDelete.yml -C // kļūdas netika atrastas

ansible-playbook -i hosts ./lu-vumc-devops.ansible/AdditionalPackagesDelete.yml // instalācija notika veiksmīgi

// reālajā dzīvē vajag pārbaudīt vai instalācija patiešam bija veiksmīga, šajā gadījumā es to pārbaudīju manuāli uz hostiem bez Ansible palīdzības

Text

Description automatically generated

Graphical user interface, text

Description automatically generated

1. Janis Sniegs atstāja Jums labošanai failu WebSuperPage.yml . Veiciet izmaiņas lai tas izpildītu savu darbu.

Hint: ieskaties:

* + Playbook name
  + Typo attieciba uz uzdevumiem
  + Json formatējums attieciba uz resursiem
  + Package Management
  + Direktoriju atļaujas serveri

nano ./lu-vumc-devops.ansible/WebSuperPage.yml // rediģēt

ansible-playbook -i hosts ./lu-vumc-devops.ansible/WebSuperPage.yml -C // testēt playbook

ansible-playbook -i ./hosts ./lu-vumc-devops.ansible/WebSuperPage.yml // instalēt playbook

ansible -i ./hosts webservers -a "curl -X GET <http://localhost/index.html>" // testēt lokālā Apache servera-darbību.

Šis uzdevums bija sarežģīts, jo Ubuntu neļāva “ pa taisno” instalēt un “startēt” Apache. Papildus nekorekti strādāja “ansible/service:” modulis un sākotnēji tas vispār neinstalēja Apache. Vajadzēja sadalīt uz 2 daļām un pēc tam ieraudzīju arī Apache masked-service problēmu.

1. Veiciet pārbaudi vai ar publisko IP Jums veras vaļa lapa. Pārbaudiet attiecīgus AWS iestādījumus.

Lapa neatvērās ar publisko IP.

Vispirms pārbaudīju vai lapa atverās no servera: ansible -i ./hosts webservers -a "curl -X GET <http://localhost/index.html>"

Pēc tam izveidoju AWS port-forward un lapa atverās arī no publiskās IP.

**\* Papildus uzdevums:** Izveidojiet un nokonfigurējiet ar Ansible struktūru, balstoties pēc šis bildes.

Diagram

Description automatically generated

Nav izpildīts.

Izpētīju pieejamās dokumentācijas, neatradu gatavu “out-of-the-box” risinājumu. Secināju, ka optimālākais ir kombinēt skriptu ar ad-hoc komandām, kas veic servera-konfigurāciju ar dažādām metodēm rediģējot konfiruācijas-failus.

Reāli neizdevās iedarbīnāt shell-komandu izpildi, zaudēju daudz laika un pazaudēju motivāciju……

ansible -i ./hosts servers -m ping

ansible -i ./hosts servers -a “sudo python3 -m http.server 8080“ &

curl -X GET http://ec2-3-139-105-82.us-east-2.compute.amazonaws.com:8080

curl -X GET http://ec2-18-220-104-220.us-east-2.compute.amazonaws.com:8080

1. Izveidojiet savu repozitoriju GitHub kur veiksiet veikto darbību aprakstu MD faila un iekopēsiet Ansible failus.

<https://github.com/maris-andersons/2grupa_maris-andersons_ansible_majasdarbs05>

Pieļāvu kļūdas darbā ar Git-repozitoriju, kā rezultātā izdzēsu daļu failu. Šajā dokumentā redzami screenshoti no failiem.

1. Pēc veiktajām darbībām, lūdzu, izdzēsiet izveidoto EC2 serveri.

Ok, status - terminated