Module B12 HW03 Project Report

Что нужно сделать

1. Создайте в Я.Облаке виртуальную машину под управлением ***Linux***. Это будет ВМ для УЦ.
2. Создайте новую внутреннюю доменную зону (используя ***Yandex Cloud DNS***), затем добавьте в эту зону новую ***DNS***-запись, указывающую на машину с ***Artifactory***. Доменное имя используйте любое на свое усмотрение.
3. На ВМ для УЦ создайте корневой и серверный сертификат для вашей машины (для выбранного вами доменного имени).
4. Настройте ***Artifatory*** на использование ***SSL*** и созданного вами сертификата. (Выбирайте любой из вариантов: настройка ***SSL*** на ***TomCat***, работающем с ***Artifactory*** веб-сервере, либо настройка связки ***nginx+Artifactory***).
5. Создайте еще одну машину в Я.Облаке, теперь под управлением ***Windows***. Она потребуется для проверки выполнения задания.
6. Установите созданный вами корневой сертификат на вашу ВМ под управлением ***Windows***. (Созданное выше доменное имя будет работать только на машинах из Я.Облака, поэтому проверить настройки с помощью своего ноутбука не выйдет).
7. С помощью веб-браузера зайдите (по ***HTTPS***-протоколу) на машину, где развернут ***Artifactory***, и убедитесь, что соединение защищено.
8. Отправьте ментору скриншот с подтверждением защищенного соединения, доменное имя вашей машины и порт, на котором работает ***Artifactory***.

Terraform.tf ca-vm

```
# ------------------------------------------------------------ VARIABLES
variable "zone" {                          # Используем переменную для передачи в конфиг инфраструктуры
  description = "Use specific availability zone" # Опционально описание переменной
  type       = string                      # Опционально тип переменной
  default    = "ru-central1-a"             # Опционально значение по умолчанию для переменной
}
variable "cloud_id" {
  type       = string                      # Опционально тип переменной
  default    = "b1gfdopk51c4d5reva85"      # Опционально значение по умолчанию для переменной
}
variable "folder_id" {
  type       = string                      # Опционально тип переменной
  default    = "b1gug0h1o834u3niipmr"      # Опционально значение по умолчанию для переменной
}
variable "cloud_key_file" {
  type       = string                      # Опционально тип переменной
  default    = "F:/DEV_HOME/Terraform_Projects/key_experiments/andrey_key.json"      # Опционально значение по умолчанию для переменной
}
variable "ssh_key_file" {
  type       = string                      # Опционально тип переменной
  default    = "F:/DEV_HOME/Terraform_Projects/key_experiments/andrey_key.pub"
}
variable "config_file" {
  type       = string                      # Опционально тип переменной
  default    = "F:/DEV_HOME/Terraform_Projects/key_experiments/andrey_config.yml"
}

# ------------------------------------------------------------ PROVIDER
terraform {
  required_providers {
    yandex = {
      source  = "yandex-cloud/yandex"
      version = "0.70.0" # Фиксируем версию провайдера
    }
  }
}

# Документация к провайдеру тут https://registry.terraform.io/providers/yandex-cloud/yandex/latest/docs#configuration-reference
# Настраиваем the Yandex.Cloud provider
provider "yandex" {
  service_account_key_file = var.cloud_key_file
  cloud_id  = var.cloud_id
  folder_id = var.folder_id
  zone      = var.zone # зона, в которая будет использована по умолчанию
}
```

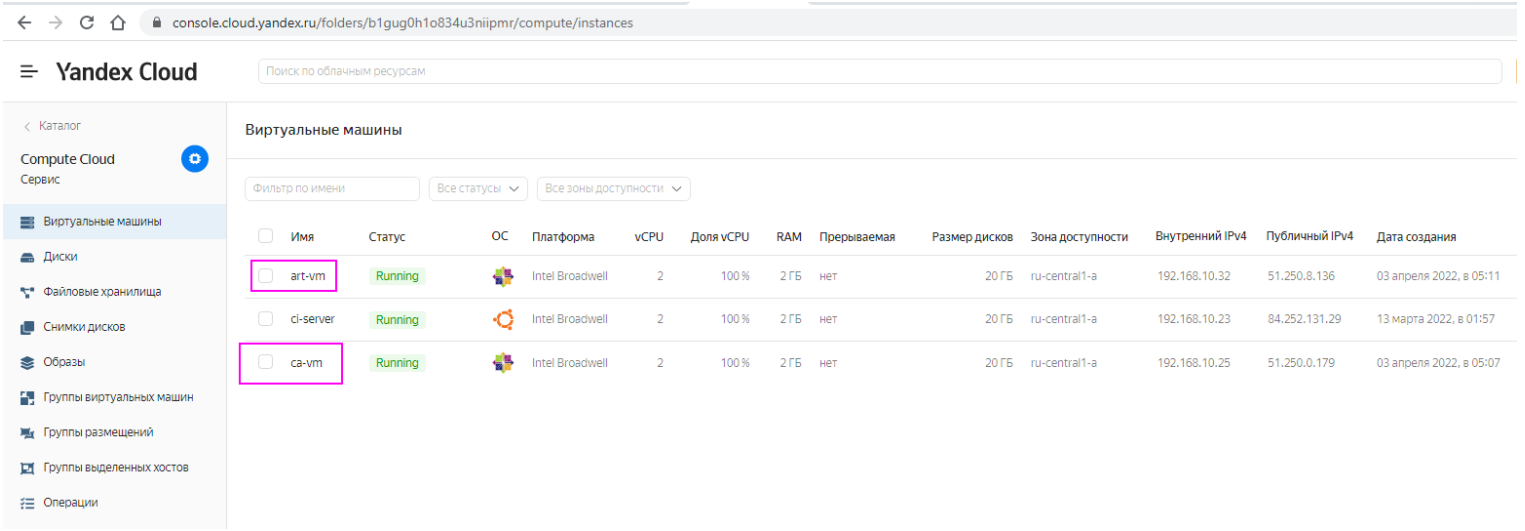
```
# ------------------------------------------------------------ WORKING CODE
data "yandex_compute_image" "centos" {
  family = "centos-7"
}

resource "yandex_compute_instance" "ca_vm" {
  name             = "ca-vm"

  resources {
    cores  = 2
    memory = 2
  }

  boot_disk {
    initialize_params {
      image_id = data.yandex_compute_image.centos.id
                 size = 20
                 type = "network-hdd"
    }
  }

  network_interface {
    subnet_id = "e9b6vh61g6p1iq9b6fe6"
    nat       = true
  }

  metadata = {
    ssh-keys = "${file(var.ssh_key_file)}"
               user-data = file(var.config_file)
  }
}


output "external_ip_address_artifact_vm" {
  value = yandex_compute_instance.ca_vm.network_interface.0.nat_ip_address
}

output "internal_ip_address_artifact_vm" {
  value = yandex_compute_instance.ca_vm.network_interface.0.ip_address
}
```

Terraform.tf art-vm

```
# ------------------------------------------------------------ VARIABLES
variable "zone" {                          # Используем переменную для передачи в конфиг инфраструктуры
  description = "Use specific availability zone" # Опционально описание переменной
  type        = string                     # Опционально тип переменной
  default     = "ru-central1-a"                # Опционально значение по умолчанию для переменной
}
variable "cloud_id" {
  type        = string                     # Опционально тип переменной
  default     = "b1gfdopk51c4d5reva85"          # Опционально значение по умолчанию для переменной
}
variable "folder_id" {
  type        = string                     # Опционально тип переменной
  default     = "b1gug0h1o834u3niipmr"          # Опционально значение по умолчанию для переменной
}
variable "cloud_key_file" {
  type        = string                     # Опционально тип переменной
  default     = "F:/DEV_HOME/Terraform_Projects/key_experiments/andrey_key.json"          # Опционально значение по умолчанию для переменной
}
variable "ssh_key_file" {
  type        = string                     # Опционально тип переменной
  default     = "F:/DEV_HOME/Terraform_Projects/key_experiments/andrey_key.pub"
}
variable "config_file" {
  type        = string                     # Опционально тип переменной
  default     = "F:/DEV_HOME/Terraform_Projects/key_experiments/andrey_config.yml"
}

# ------------------------------------------------------------ PROVIDER
terraform {
```

```
  required_providers {
    yandex = {
      source  = "yandex-cloud/yandex"
      version = "0.70.0" # Фиксируем версию провайдера
    }
  }
}

# Документация к провайдеру тут https://registry.terraform.io/providers/yandex-cloud/yandex/latest/docs#configuration-reference
# Настраиваем the Yandex.Cloud provider
provider "yandex" {
  service_account_key_file = var.cloud_key_file
  cloud_id  = var.cloud_id
  folder_id = var.folder_id
  zone      = var.zone # зона, в которая будет использована по умолчанию
}

# ------------------------------------------------------------ WORKING CODE
data "yandex_compute_image" "centos" {
  family = "centos-7"
}

resource "yandex_compute_instance" "artifact_vm" {
  name              = "art-vm"

  resources {
    cores  = 2
    memory = 2
  }

  boot_disk {
    initialize_params {
      image_id = data.yandex_compute_image.centos.id
                 size = 20
                 type = "network-hdd"
    }
  }

  network_interface {
    subnet_id = "e9b6vh61g6p1iq9b6fe6"
    nat       = true
  }

  metadata = {
    ssh-keys = "${file(var.ssh_key_file)}"
               user-data = file(var.config_file)
  }
}


output "external_ip_address_artifact_vm" {
  value = yandex_compute_instance.artifact_vm.network_interface.0.nat_ip_address
}

output "internal_ip_address_artifact_vm" {
  value = yandex_compute_instance.artifact_vm.network_interface.0.ip_address
}
```

Result



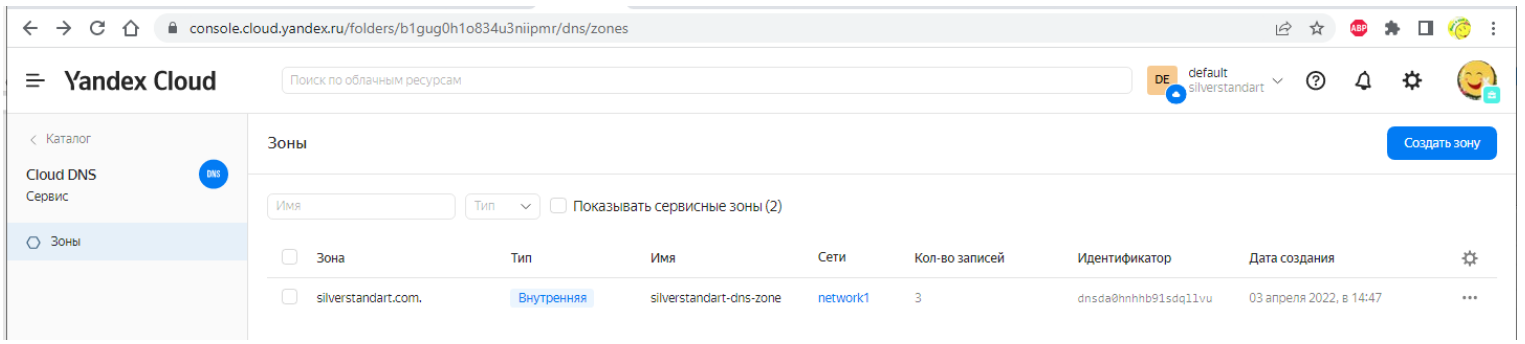
---------- certification assign server ca-vm
external_ip_address_artifact_vm = "51.250.0.179"
internal_ip_address_artifact_vm = "192.168.10.25"

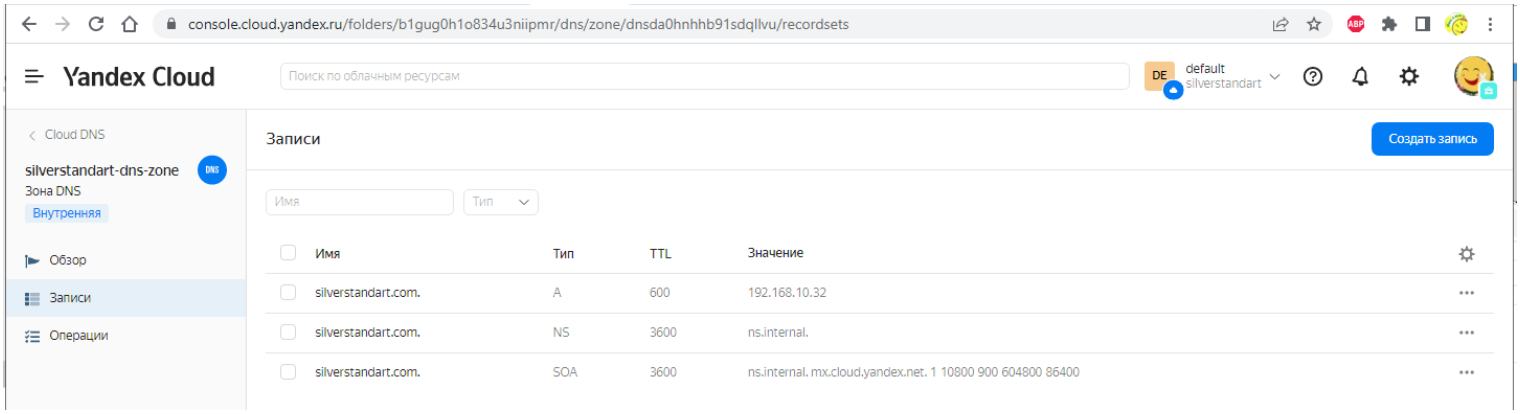
ssh.exe -i F:/DEV_HOME/Terraform_Projects/key_experiments/andrey_key andrey@51.250.0.179

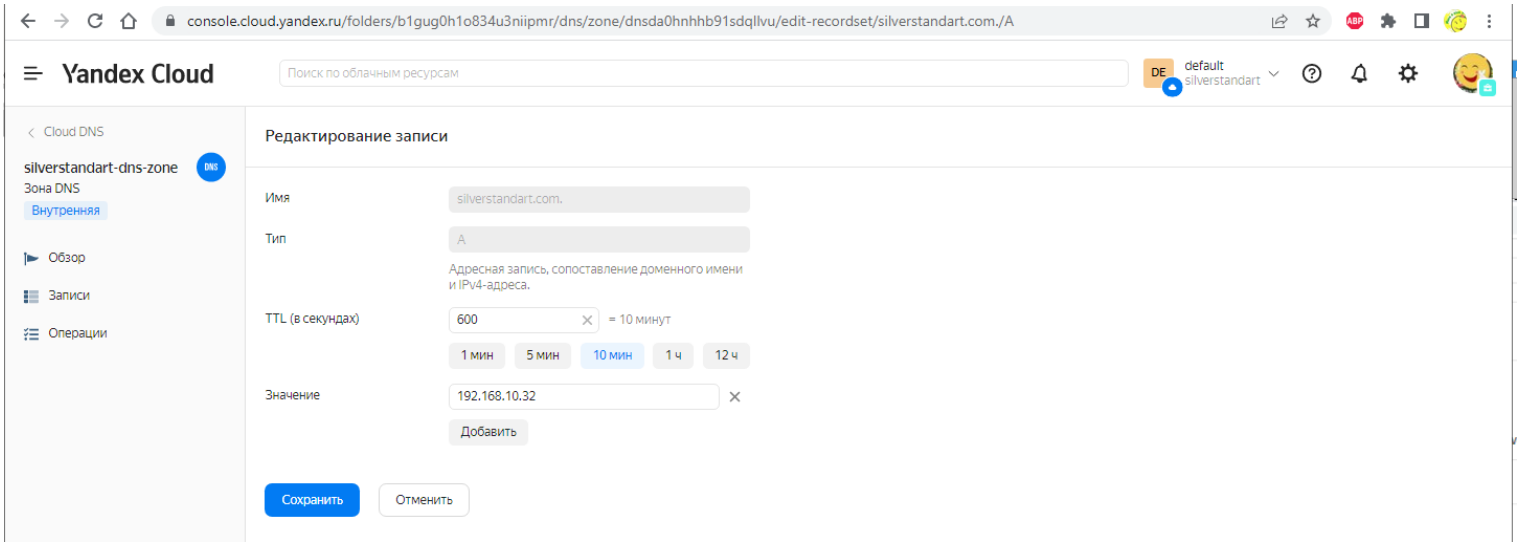
---------- Artifact server NGINX art-vm
external_ip_address_artifact_vm = "51.250.8.136"
internal_ip_address_artifact_vm = "192.168.10.32"

ssh.exe -i F:/DEV_HOME/Terraform_Projects/key_experiments/andrey_key andrey@51.250.8.136

Create domain name



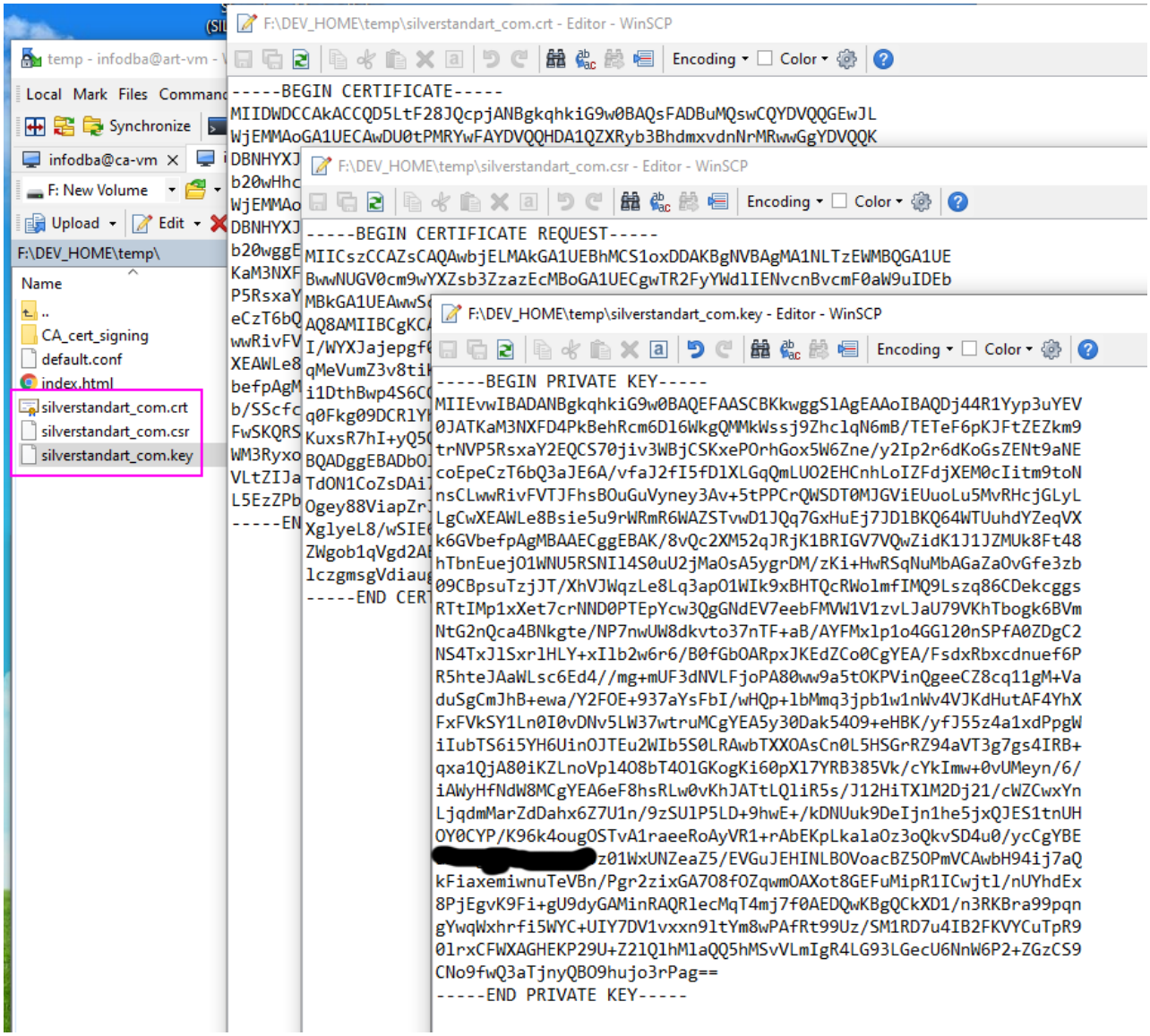




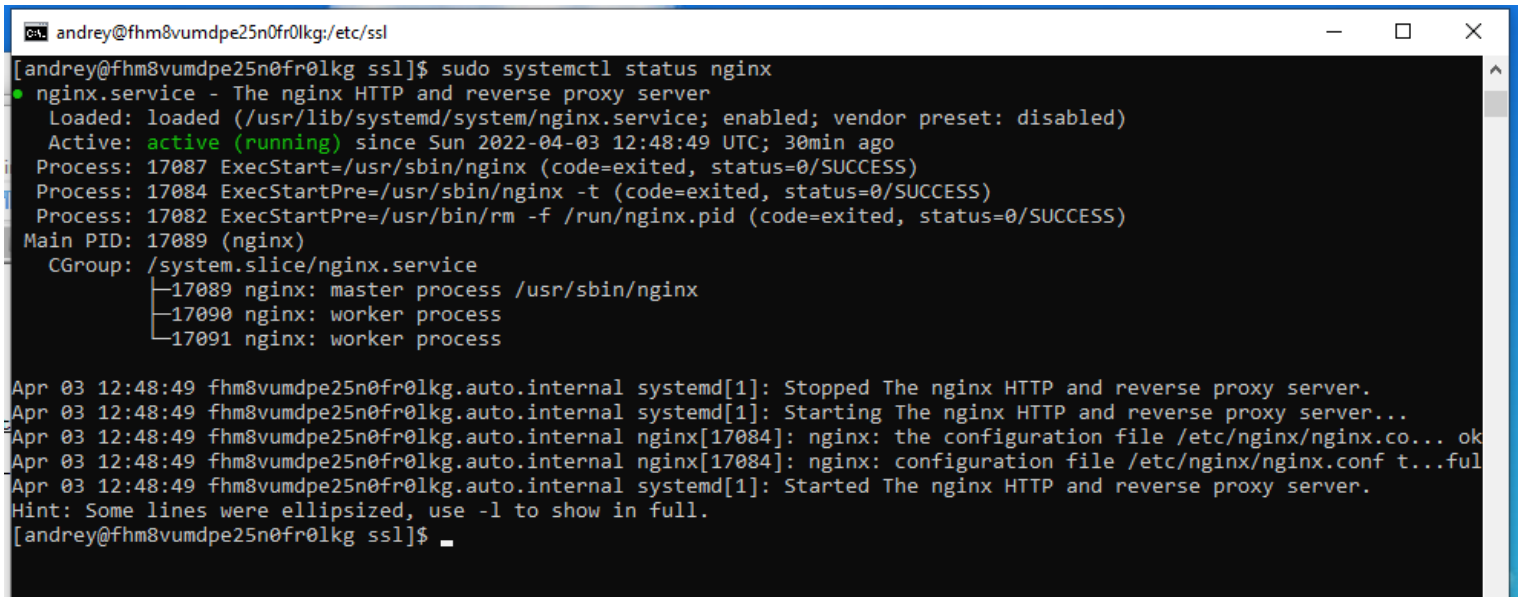
Certification generation in **ca-vm**

```
# ---------  openssl  https://www.digitalocean.com/community/tutorials/openssl-essentials-working-with-ssl-certificates-private-keys-and-csrs
openssl req -subj "/C=KZ/ST=SKO/L=Petropavlovsk/O=Garage Corporation /CN=silverstandart.com" \
    -newkey rsa:2048 -nodes -keyout silverstandart_com.key \
    -out silverstandart_com.csr

openssl x509 \
    -signkey silverstandart_com.key \
    -in silverstandart_com.csr \
    -req -days 365 -out silverstandart_com.crt
```

```
[andrey@fhmnvfckiu15k7ehstko ssl]$ openssl rsa -noout -modulus -in silverstandart_com.key | openssl md5
(stdin)= cd2861ef95a5584bdca255b3d11836c2
[andrey@fhmnvfckiu15k7ehstko ssl]$ openssl x509 -noout -modulus -in silverstandart_com.crt | openssl md5
(stdin)= cd2861ef95a5584bdca255b3d11836c2
[andrey@fhmnvfckiu15k7ehstko ssl]$ openssl req -noout -modulus -in silverstandart_com.csr | openssl md5
(stdin)= cd2861ef95a5584bdca255b3d11836c2
[andrey@fhmnvfckiu15k7ehstko ssl]$
```



Nginx running and ports



```
andrey@fhm8vumdpe25n0fr0lkg:/etc/ssl
[andrey@fhm8vumdpe25n0fr0lkg ssl]$ netstat -tulnp
(No info could be read for "-p": geteuid()=1000 but you should be root.)
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State       PID/Program name
tcp        0      0 0.0.0.0:111             0.0.0.0:*               LISTEN      -
tcp        0      0 0.0.0.0:80              0.0.0.0:*               LISTEN      -
tcp        0      0 0.0.0.0:22              0.0.0.0:*               LISTEN      -
tcp        0      0 127.0.0.1:25            0.0.0.0:*               LISTEN      -
tcp        0      0 0.0.0.0:443             0.0.0.0:*               LISTEN      -
tcp6       0      0 :::111                  :::*                    LISTEN      -
tcp6       0      0 :::80                   :::*                    LISTEN      -
tcp6       0      0 :::22                   :::*                    LISTEN      -
tcp6       0      0 ::1:25                  :::*                    LISTEN      -
udp        0      0 0.0.0.0:68              0.0.0.0:*                           -
udp        0      0 0.0.0.0:111             0.0.0.0:*                           -
udp        0      0 127.0.0.1:323           0.0.0.0:*                           -
udp        0      0 0.0.0.0:731             0.0.0.0:*                           -
udp6       0      0 :::111                  :::*                                -
udp6       0      0 ::1:323                 :::*                                -
udp6       0      0 :::731                  :::*                                -
[andrey@fhm8vumdpe25n0fr0lkg ssl]$
```

Result check

http + domain

```
andrey@fhmnvfckiu15k7ehstko:~/ssl
[andrey@fhmnvfckiu15k7ehstko ssl]$ curl http://silverstandart.com
<!DOCTYPE html>
<html>
<body>

<h1>My First Page</h1>
<p>This page created by Andrey as a part of B12_HW03</p>

</body>
</html>

[andrey@fhmnvfckiu15k7ehstko ssl]$
```

```
andrey@fhmnvfckiu15k7ehstko:~/ssl
[andrey@fhmnvfckiu15k7ehstko ssl]$ wget http://silverstandart.com
--2022-04-03 13:27:06--  http://silverstandart.com/
Resolving silverstandart.com (silverstandart.com)... 192.168.10.32
Connecting to silverstandart.com (silverstandart.com)|192.168.10.32|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 139 [text/html]
Saving to: ‘index.html.6’

100%[======================================================================================>] 139         --.-K/s   in 0s

2022-04-03 13:27:06 (20.0 MB/s) - ‘index.html.6’ saved [139/139]

[andrey@fhmnvfckiu15k7ehstko ssl]$
```

SSL connection

```
andrey@fhmnvfckiu15k7ehstko:~/ssl
[andrey@fhmnvfckiu15k7ehstko ssl]$ wget https://silverstandart.com
--2022-04-03 13:27:50--  https://silverstandart.com/
Resolving silverstandart.com (silverstandart.com)... 192.168.10.32
Connecting to silverstandart.com (silverstandart.com)|192.168.10.32|:443... connected.
ERROR: cannot verify silverstandart.com's certificate, issued by ‘/C=KZ/ST=SKO/L=Petropavlovsk/O=Garage Corporation /CN=
silverstandart.com’:
  Self-signed certificate encountered.
To connect to silverstandart.com insecurely, use `--no-check-certificate'.
[andrey@fhmnvfckiu15k7ehstko ssl]$
```

Ignore cert check