

# Module B12 HW03 Project Report

## Что нужно сделать

1. Создайте в Я.Облаке виртуальную машину под управлением **Linux**. Это будет VM 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 VM для УЦ создайте корневой и серверный сертификат для вашей машины (для выбранного вами доменного имени).
4. Настройте **Artifactory** на использование **SSL** и созданного вами сертификата. (Выбирайте любой из вариантов: настройка **SSL** на **TomCat**, работающем с **Artifactory** веб-сервере, либо настройка связи **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 VM 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" {
  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

← → ↺ 🏠

console.cloud.yandex.ru/folders/b1gug0h1o834u3niipmr/compute/instances

☰ Yandex Cloud

Поиск по облачным ресурсам

Каталог

Compute Cloud Сервис

Виртуальные машины

Диски

Файловые хранилища

Снимки дисков

Образы

Группы виртуальных машин

Группы размещений

Группы выделенных хостов

Операции

Виртуальные машины

Фильтр по имени Все статусы Все зоны доступности

<input type="checkbox"/>	Имя	Статус	ОС	Платформа	vCPU	Доля vCPU	RAM	Прерываемая	Размер дисков	Зона доступности	Внутренний IPv4	Публичный IPv4	Дата создания
<input type="checkbox"/>	art-vm	Running		Intel Broadwell	2	100 %	2 ГБ	нет	20 ГБ	ru-central1-a	192.168.10.32	51.250.8.136	03 апреля 2022, в 05:11
<input type="checkbox"/>	ci-server	Running		Intel Broadwell	2	100 %	2 ГБ	нет	20 ГБ	ru-central1-a	192.168.10.23	84.252.131.29	13 марта 2022, в 01:57
<input type="checkbox"/>	ca-vm	Running		Intel Broadwell	2	100 %	2 ГБ	нет	20 ГБ	ru-central1-a	192.168.10.25	51.250.0.179	03 апреля 2022, в 05:07

----- 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

← → ↺ 🏠

console.cloud.yandex.ru/folders/b1gug0h1o834u3niipmr/dns/zones

☰ Yandex Cloud

Поиск по облачным ресурсам

Каталог

Cloud DNS Сервис

Зоны

Зоны

Имя Тип ☐ Показывать сервисные зоны (2)

Создать зону

<input type="checkbox"/>	Зона	Тип	Имя	Сети	Кол-во записей	Идентификатор	Дата создания	
<input type="checkbox"/>	silverstandart.com.	Внутренняя	silverstandart-dns-zone	network1	3	dnsda0hnhhb91sdq11vu	03 апреля 2022, в 14:47	...

← → ↺ 🏠

console.cloud.yandex.ru/folders/b1gug0h1o834u3niipmr/dns/zone/dnsda0hnhhb91sdq11vu/recordsets

☰ Yandex Cloud

Поиск по облачным ресурсам

Cloud DNS

silverstandart-dns-zone Зона DNS Внутренняя

Обзор

Записи

Операции

Записи

Имя Тип

Создать запись

<input type="checkbox"/>	Имя	Тип	TTL	Значение	
<input type="checkbox"/>	silverstandart.com.	A	600	192.168.10.32	...
<input type="checkbox"/>	silverstandart.com.	NS	3600	ns.internal.	...
<input type="checkbox"/>	silverstandart.com.	SOA	3600	ns.internal. mx.cloud.yandex.net. 1 10800 900 604800 86400	...

← → ↺ 🏠 console.cloud.yandex.ru/folders/b1gug0h1o834u3niipmr/dns/zone/dnsda0hnhhb91sdqllvu/edit-recordset/silverstandart.com./A

Yandex Cloud

default silverstandart

< Cloud DNS

silverstandart-dns-zone DNS

Зона DNS

Внутренняя

Обзор

Записи

Операции

### Редактирование записи

Имя

Тип

Адресная запись, сопоставление доменного имени и IPv4-адреса.

TTL (в секундах)  × = 10 минут

1 мин 5 мин 10 мин 1 ч 12 ч

Значение  ×

Добавить

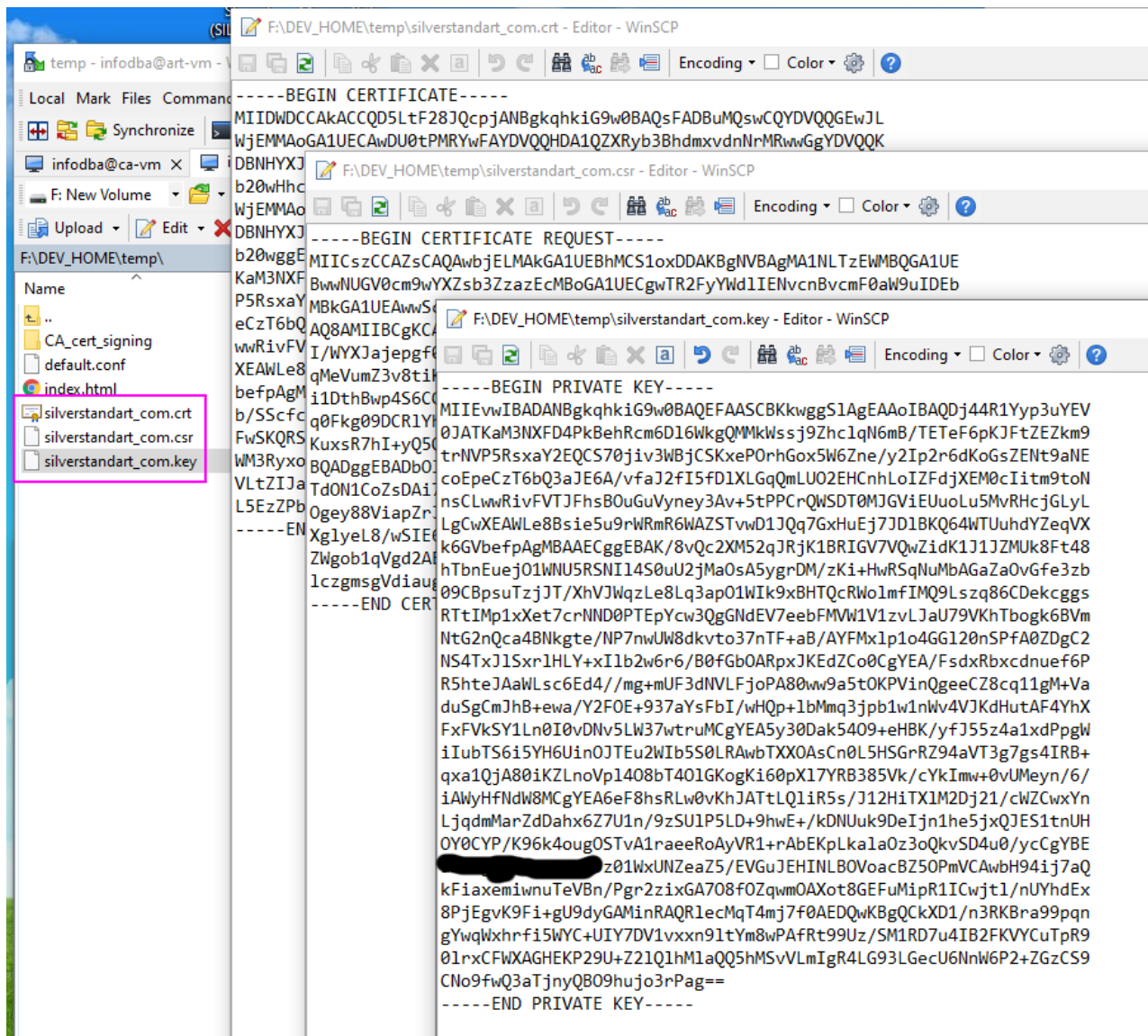
Сохранить Отменить

## 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=Petrovsk/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@fhmvmfckiu15k7ehstko ssl]$ openssl rsa -noout -modulus -in silverstandart_com.key | openssl md5
(stdin)= cd2861ef95a5584bdca255b3d11836c2
[andrey@fhmvmfckiu15k7ehstko ssl]$ openssl x509 -noout -modulus -in silverstandart_com.crt | openssl md5
(stdin)= cd2861ef95a5584bdca255b3d11836c2
[andrey@fhmvmfckiu15k7ehstko ssl]$ openssl req -noout -modulus -in silverstandart_com.csr | openssl md5
(stdin)= cd2861ef95a5584bdca255b3d11836c2
[andrey@fhmvmfckiu15k7ehstko ssl]$
```



## Nginx running and ports

```

andrey@fhm8vumdpe25n0fr0lkg:/etc/ssl
[andrey@fhm8vumdpe25n0fr0lkg ssl]$ sudo systemctl status nginx
● nginx.service - The nginx HTTP and reverse proxy server
   Loaded: loaded (/usr/lib/systemd/system/nginx.service; enabled; vendor preset: disabled)
   Active: active (running) since Sun 2022-04-03 12:48:49 UTC; 30min ago
     Process: 17087 ExecStart=/usr/sbin/nginx (code=exited, status=0/SUCCESS)
     Process: 17084 ExecStartPre=/usr/sbin/nginx -t (code=exited, status=0/SUCCESS)
     Process: 17082 ExecStartPre=/usr/bin/rm -f /run/nginx.pid (code=exited, status=0/SUCCESS)
   Main PID: 17089 (nginx)
   CGroup: /system.slice/nginx.service
           └─17089 nginx: master process /usr/sbin/nginx
             └─17090 nginx: worker process
               └─17091 nginx: worker process

Apr 03 12:48:49 fhm8vumdpe25n0fr0lkg.auto.internal systemd[1]: Stopped The nginx HTTP and reverse proxy server.
Apr 03 12:48:49 fhm8vumdpe25n0fr0lkg.auto.internal systemd[1]: Starting The nginx HTTP and reverse proxy server...
Apr 03 12:48:49 fhm8vumdpe25n0fr0lkg.auto.internal nginx[17084]: nginx: the configuration file /etc/nginx/nginx.conf... ok
Apr 03 12:48:49 fhm8vumdpe25n0fr0lkg.auto.internal nginx[17084]: nginx: configuration file /etc/nginx/nginx.conf t...ful
Apr 03 12:48:49 fhm8vumdpe25n0fr0lkg.auto.internal systemd[1]: Started The nginx HTTP and reverse proxy server.
Hint: Some lines were ellipsized, use -l to show in full.
[andrey@fhm8vumdpe25n0fr0lkg ssl]$

```

```
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@fhm8vumdpe25n0fr0lkg:~/ssl
[andrey@fhm8vumdpe25n0fr0lkg 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@fhm8vumdpe25n0fr0lkg ssl]$
```

```
andrey@fhm8vumdpe25n0fr0lkg:~/ssl
[andrey@fhm8vumdpe25n0fr0lkg 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@fhm8vumdpe25n0fr0lkg ssl]$
```

SSL connection

```
andrey@fhm8vumdpe25n0fr0lkg:~/ssl
[andrey@fhm8vumdpe25n0fr0lkg 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@fhm8vumdpe25n0fr0lkg ssl]$
```



Ignore cert check

```
andrey@fhmnvfckiu15k7ehstko:~/ssl
[andrey@fhmnvfckiu15k7ehstko ssl]$ wget https://silverstandart.com --no-check-certificate
--2022-04-03 13:28:40-- https://silverstandart.com/
Resolving silverstandart.com (silverstandart.com)... 192.168.10.32
Connecting to silverstandart.com (silverstandart.com)[192.168.10.32]:443... connected.
WARNING: cannot verify silverstandart.com's certificate, issued by '/C=KZ/ST=SKO/L=Petrodavlovsk/O=Garage Corporation /CN=silverstandart.com':
  Self-signed certificate encountered.
HTTP request sent, awaiting response... 200 OK
Length: 4833 (4.7K) [text/html]
Saving to: 'index.html.8'

100%[=====>] 4,833      ---K/s   in 0s

2022-04-03 13:28:40 (606 MB/s) - 'index.html.8' saved [4833/4833]

[andrey@fhmnvfckiu15k7ehstko ssl]$
```