

Инструкция по установке
программного обеспечения для
электронно-вычислительных машин
Arenadata One (AOne)

Оглавление

1	Введение	3
2	Создание инфраструктуры в облаке VK cloud	3
2.1	Шаг 1. Предварительные требования	4
2.1.1	Требования к software и hardware	4
2.1.2	Установка Ansible	4
2.1.3	Установка Terraform	5
2.1.4	Установка Helm	5
2.1.5	Скачивание исходного кода проекта	5
2.2	Шаг 2. Развертывание облачного k8s кластера и экземпляра FreeIpa	6
2.3	Шаг 3. Проверка установки	9
2.3.1	Проверка доступности freeIPA	9
2.3.2	Проверка доступности кластера	11
2.3.3	Распечатывание Vault	12
2.3.4	Расшифровка паролей	14
2.4	Шаг 4. Установка и конфигурирование инфраструктурных компонентов	14
2.4.1	Подготовка окружения	14
2.4.2	Конфигурирование Vault	15
2.4.3	Установка certificate-manager	15
2.4.4	Установка инфраструктурных компонентов	15
2.4.5	Установка дополнительных инфраструктурных компонентов	23
2.5	Шаг 5. Установка приложений	23

1 Введение

Инструкция посвящена online-установке кластера Arenadata One (AOne). Данный способ установки подходит для хостов с доступом к Интернету и внутренним сервисам Arenadata (Gitlab и harbor).

Процесс online-установки с помощью terraform и ansible включает следующие шаги:

- Создание инфраструктуры в облаке VK cloud;
- Развертывание облачного k8s кластера и экземпляра Freelpa;
- Установка инфраструктурных компонентов в кластер;
- Установка сервисных компонентов в кластер.

2 Создание инфраструктуры в облаке VK cloud

Для развертывания инфраструктуры в облаке VK cloud требуется доступ в личный кабинет и активированный «Доступ по API» для пользователя.

Активация доступа по API необходима для работы с VK Cloud при помощи публичного API и дополнительных инструментов, таких как OpenStack CLI, Terraform, kubectl и другие.

Для активации доступа по API:

- Включить двухфакторную аутентификацию;
- Нажать на имя пользователя в шапке страницы и включить доступ по API одним из способов:

Настройки проекта:

- Из выпадающего списка выбрать **Настройки проекта**;
- На вкладке **Доступ по API** нажать **Активировать доступ по API**.

Настройки аккаунта:

- Из выпадающего списка выбрать **Настройки аккаунта**;
- Перейти в раздел **Безопасность аккаунта VK Cloud**;
- В блоке **Дополнительные доступы** включить опцию **Доступ по API**.

Для деактивации доступа по API:

- Включить двухфакторную аутентификацию, если она была отключена;
- Нажать на имя пользователя в шапке страницы и выбрать **Настройки аккаунта**;
- В разделе **Безопасность** отключить опцию **Доступ по API**.

2.1 Шаг 1. Предварительные требования

2.1.1 Требования к software и hardware

AOne поставляется в виде набора terraform манифестов и Anisble скриптов.

Для его установки необходимы:

- Установленные программные продукты Terraform ($\geq 1.10.4$), Ansible (не выше версии 2.16.14), Helm;
- Пользователь с правами sudo;
- Доступ к официальным репозиториям terraform. По умолчанию, все репозитории доступны в РФ.

Также перед началом установки необходимо отредактировать файл `~/.ssh/config`, чтобы впоследствии пользователь, выполняющий установку AOne, смог управлять созданной инфраструктурой. Для этого выполнить команду:

```
sudo vi ~/.ssh/config
```

В этом файле необходимо добавить строчки:

```
Host *  
  AddKeysToAgent yes
```

Чтобы сохранить это изменение и вернуться в терминал, нажать **Esc**, после чего набрать `:wq` либо `:wq!`.

2.1.2 Установка Ansible

Устанавливать Ansible рекомендуется с официального сайта.

Версия Ansible не должна быть выше **v2.16.14**. Начиная в версии **v2.17** Ansible прокидывает `ruenv` в `host`, что ломает деплой `freeipa`.

Для mac возможно использовать пакетный менеджер brew, например: `brew install ansible@9`

2.1.3 Установка Terraform

Устанавливать Terraform рекомендуется с официального сайта.

- Linux CentOS:

```
sudo yum install -y yum-utils sudo yum-config-manager --add-repo  
https://rpm.releases.hashicorp.com/RHEL/hashicorp.repo sudo yum -y install  
terraform
```

- Mac:

```
brew tap hashicorp/tap  
brew install hashicorp/tap/terraform
```

2.1.4 Установка Helm

Устанавливать Terraform рекомендуется с официального сайта.

- Linux:

```
curl -fsSL -o get_helm.sh  
https://raw.githubusercontent.com/helm/helm/main/scripts/get-helm-3 chmod  
700 get_helm.sh ./get_helm.sh
```

- Mac:

```
brew install helm  
brew install helmfile
```

2.1.5 Скачивание исходного кода проекта

Необходимо скачать исходный код проекта из Git репозитория Arenadata GitLab:

```
sudo git clone -b aaf/noTLS  
https://gitlab.adsw.io/arenadata/ng/platform/helmfiles.git
```

На данный момент нет возможности динамически выделять доменные имена для кластеров и продуктов, для тестового деплоя рекомендуется ставить noTLS версию кластера, где отключены шаги создания сертификатов для Ingress controller. Данная версия хранится в ветке *aaf/noTLS*. Перед деплоем необходимо перейти в данную ветку командой `git checkout aaf/noTLS`.

2.2 Шаг 2. Развертывание облачного k8s кластера и экземпляра FreeIPA

- 1). Установить все переменные для кластера и хоста **freeipa** в зависимости от требований в *infrastructure/variables.tf*.

В репозитории хранится подготовленный файл *variables.tf*, который развернет в облачной инфраструктуре VK Cloud k8s кластер с именем *adone-test* и экземпляр freeIPA с доменным именем *freeipa.test*. В продуктовой среде рекомендуется использовать реальные DNS имена.

Пример заполненного файла *infrastructure/variables.tf*:

```
variable "login" {
  default = "login"
}
variable "password" {
  default = "password"
}
variable "project_id" {
  default = "project_id"
}
variable "token" {
  default = "token"
}
variable "cluster_name" {
  default = "adone-test"
}
variable "master_count" {
  default = 1
}
variable "master_flavor" {
  default = "STD2-4-16"
}
variable "node_count" {
  default = 8
}
variable "node_flavor" {
  default = "STD2-8-32"
}
variable "volume_node_size" {
  default = 50
}
variable "volume_node_type" {
  default = "ceph-ssd"
}
variable "volume_master_size" {
  default = 50
}
variable "volume_master_type" {
  default = "ceph-ssd"
}
```

```
variable "av_zone" {
    default = "ME1"
}
variable "network_name" {
    default = "k8s-test"
}
variable "subnet_name" {
    default = "k8s-subnet-test"
}
variable "cidr" {
    default = "192.168.198.0/24"
}
##### freeipa #####

variable "hostname" {
    default = "freeipa-test"
}
variable "freeipa_flavor" {
    default = "STD2-2-2"
}
variable "freeipa_root_volume_size" {
    default = "20"
}
variable "freeipa_root_volume_type" {
    default = "ceph-ssd"
}
variable "path_to_ansible" {
    default = "./ansible"
}
variable "cn_name" {
    default = "freeipa.test"
}
variable "ansible_user" {
    default = "centos"
}
variable "ipaadmin_password" {
    default = "SparkLight2024!"
}
variable "ipadm_password" {
    default = "password"
}
variable "ipaserver_domain" {
    default = "freeipa.test"
}
variable "ipaserver_realm" {
    default = "freeipa.test"
}
variable "ipaserver_setup_dns" {
    description = "choose yes or no"
    default = "no"
}
variable "ipaserver_auto_forwarders" {
    description = "choose yes or no"
```

```
    default = "no"
  }
  variable "freeipa_repository_path" {
    default = "https://github.com/freeipa/ansible-freeipa.git"
  }

  variable "does_network_exist" {
    description = <<EOF
      We need to choose which network we want to use.
      If network already exists we need to set true.
      If we need to create new network infrastructure set to false
    EOF
    type = bool
    default = true
  }

  variable "exist_network_name" {
    description = "If we choose existing network. Set its name"
    default = "nextgen"
  }

  variable "exist_subnet_name" {
    description = "If we choose existing network. Set subnet name"
    default = "nextgen_subnet_1"
  }

  variable "does_key_exist" {
    type = bool
    default = false
  }

  variable "floating_ip_enabled" {
    type = bool
    default = false
  }
}
```

2). Переименовать *infrastructure/terraform.tfvars.simple* в *infrastructure/terraform.tfvars* и заполнить учетные данные.

3). Выполнить команду `terraform init` для установки необходимых terraform провайдеров.

Результат выполнения команды:

```
terraform init

Initializing the backend...
Initializing provider plugins...
- Finding vk-cs/vkcs versions matching "~> 0.7.4"...
- Finding mumoshu/helmfile versions matching "0.14.1"...
- Finding latest version of hashicorp/local...
- Finding latest version of hashicorp/null...
- Finding latest version of hashicorp/template...
```



```
- Installing vk-cs/vkcs v0.7.4...
- Installed vk-cs/vkcs v0.7.4 (self-signed, key ID 446513894982E1EA)
- Installing mumoshu/helmfile v0.14.1...
- Installed mumoshu/helmfile v0.14.1 (unauthenticated)
- Installing hashicorp/local v2.5.2...
- Installed hashicorp/local v2.5.2 (signed by HashiCorp)
- Installing hashicorp/null v3.2.3...
- Installed hashicorp/null v3.2.3 (signed by HashiCorp)
- Installing hashicorp/template v2.2.0...
- Installed hashicorp/template v2.2.0 (unauthenticated)
Partner and community providers are signed by their developers.
If you'd like to know more about provider signing, you can read about it
here:
https://www.terraform.io/docs/cli/plugins/signing.html
Terraform has created a lock file .terraform.lock.hcl to record the provider
selections it made above. Include this file in your version control
repository
so that Terraform can guarantee to make the same selections by default when
you run "terraform init" in the future.
```

Для устранения ошибки о несовместимости провайдера и платформы:

```
| Error: Incompatible provider version
|
| Provider registry.terraform.io/hashicorp/template v2.2.0 does not have a
package available for your current platform, darwin_arm64.
|
| Provider releases are separate from Terraform CLI releases, so not all
providers are available for all platforms. Other versions of this provider
| may have different platforms supported.
```

Необходимо использовать данный workaround:

```
brew install kreuzwerker/taps/ml-terraform-provider-helper
ml-terraform-provider-helper activate
```

4). Выполнить команду `terraform apply` внутри каталога «*infrastructure*» для создания k8s кластера и экземпляра FreeIPA.

2.3 Шаг 3. Проверка установки

2.3.1 Проверка доступности freeIPA

После создания кластера и экземпляра freeIPA появятся новые файлы в каталоге *etc_hosts* и *freeipa_keys*. Здесь находятся данные для редактирования */etc/hosts* и ключи для доступа к серверу freeIPA. Также появится новая папка - *infrastructure/kubecofnig* с файлом *kubecofnig* для доступа к k8s кластеру.

В процессе создания и конфигурирования экземпляра freeIPA в аккаунте создается SSH ключевая пара *freeipa_access_key*, значение которой будет добавлено в *ssh-agent* для дальнейшего конфигурирования экземпляра freeIPA.

Для проверки конфигурирования экземпляра free-ipa необходимо проверить наличие ключа из файла *freeipa_keys* в выводе команды `ssh-add -L`. А также возможность подключения в виртуальной машине `ssh ip -lcentos`. Также при редактировании файла */etc/hosts* появляется возможность использования UI freeIPA. Для проверки этого выполнить команду:

```
curl -L -k freeipa-test.freeipa.test
```

Результат команды:

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8">
  <title>Identity Management</title>

  <!--[if IE]>
  <meta id="ie-detector">
  <![endif]-->

  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <script type="text/javascript" src="js/libs/loader.js"></script>
  <script type="text/javascript">

    var dojoConfig = {
      baseUrl: "js",
      has: {
        'dojo-firebug': false,
        'dojo-debug-messages': true
      },
      parseOnLoad: false,
      async: true,
      packages: [
        {
          name: 'libs',
          location: 'libs'
        },
        {
          name: 'dojo',
          location: 'dojo'
        },
        {
          name: 'freeipa',
          location: 'freeipa'
        }
      ]
    };
  </script>
</head>
</html>
```

```
    ],
    cacheBust: ipa_loader.num_version || ""
  };

  (function() {
    var ie = !!document.getElementById('ie-detector');
    var styles = [
      'css/patternfly.css',
      'css/bootstrap-datepicker3.min.css',
      'css/ipa.css',
      'ipa.css'
    ];
    if (ie) styles.push('ie.css');
    var icons = ['favicon.ico'];
    var scripts = [
      'js/libs/json2.js',
      'js/libs/jquery.js',
      'js/libs/bootstrap.js',
      'js/libs/bootstrap-datepicker.js',
      'js/libs/patternfly.js',
      'js/libs/jquery.ordered-map.js',
      'js/libs/browser.js',
      'js/dojo/dojo.js',
      'js/libs/qrcode.js'
    ];
    ipa_loader.scripts(scripts, function() {
      require(['freeipa/app'], function(app) { app.run(); });
    });
    ipa_loader.styles(styles);
    ipa_loader.icons(icons);

  })();
</script>
</head>

<body>
  <noscript>This application requires JavaScript enabled.</noscript>
</body>

</html>%
```

2.3.2 Проверка доступности кластера

При установке кластера AOne автоматически создается кластер K8s. Для управления кластером, необходимо сконфигурировать **kubecfg** в директории *.kube*. Для этого необходимо скопировать содержимое файла из *infrastructure/kubecfg* в файл *.kube/config*. После данной операции можно управлять k8s кластером.

Для проверки следует выполнить команду:

```
kubectl get nodes
```

Результат команды:

NAME	STATUS	ROLES	AGE	VERSION
adone-test-adone-test-node-group-0	Ready	<none>	151m	v1.27.6
adone-test-adone-test-node-group-1	Ready	<none>	151m	v1.27.6
adone-test-adone-test-node-group-2	Ready	<none>	150m	v1.27.6
adone-test-adone-test-node-group-3	Ready	<none>	151m	v1.27.6
adone-test-adone-test-node-group-4	Ready	<none>	151m	v1.27.6
adone-test-adone-test-node-group-5	Ready	<none>	150m	v1.27.6
adone-test-adone-test-node-group-6	Ready	<none>	151m	v1.27.6
adone-test-adone-test-node-group-7	Ready	<none>	151m	v1.27.6
adone-test-master-0	Ready	master	157m	v1.27.6

2.3.3 Распечатывание Vault

При установке кластера AOne автоматически устанавливается Vault.

Изначально vault запечатан, чтобы распечатать vault, необходимо выполнить следующую команду:

```
kubectl get pods -n vault
```

Результат команды:

```
NAME READY STATUS RESTARTS AGE vault-injector-85649787cb-p27vh 1/1 Running 0  
150m vault-server-0 0/1 Running 0 150m
```

Необходимо подключиться к контейнеру с именем *vault-server-0* и распечатать vault, для этого выполнить следующую команду:

```
kubectl -n vault exec -it vault-server-0 -- bash  
vault operator init  
vault operator unseal
```

Результат команды:

```
vault@vault-server-0:/$ vault operator init  
Unseal Key 1: 93HUJDQVkuLLjeSyUJLTEUA5ZKo1Vuublrvl877/A+F8  
Unseal Key 2: b7CGuc4ASqoEZ4kwQR0Wmdpffjvlq95UsGujSwa9sRO6c  
Unseal Key 3: cDgjCIly5iHpTYMGXmOK6KMB1owxz3jEkDBC9dGFEnZS  
Unseal Key 4: M4QNoqSBThaSO5J7OsdVXIbezI175e8i4bLMHY0koHS1  
Unseal Key 5: VkP1x/o+Qc/XHovT9w7y9AzbfIy1gNGJ9VjCOuojtq11  
  
Initial Root Token: hvs.qnwhx44OUBwLWxlwiPfc1se8  
  
Vault initialized with 5 key shares and a key threshold of 3. Please  
securely  
distribute the key shares printed above. When the Vault is re-sealed,  
restarted, or stopped, you must supply at least 3 of these keys to unseal it
```

before it can start servicing requests.

Vault does not store the generated root key. Without at least 3 keys to reconstruct the root key, Vault will remain permanently sealed!

It is possible to generate new unseal keys, provided you have a quorum of existing unseal keys shares. See "vault operator rekey" for more information.

```
vault@vault-server-0:/$ vault operator unseal
```

```
Unseal Key (will be hidden):
```

Key	Value
---	-----
Seal Type	shamir
Initialized	true
Sealed	true
Total Shares	5
Threshold	3
Unseal Progress	1/3
Unseal Nonce	5f6ef075-ff8e-dacc-4bbf-ff51901ed729
Version	1.16.2
Build Date	2024-04-22T16:25:54Z
Storage Type	raft
HA Enabled	true

```
vault@vault-server-0:/$ vault operator unseal
```

```
Unseal Key (will be hidden):
```

Key	Value
---	-----
Seal Type	shamir
Initialized	true
Sealed	true
Total Shares	5
Threshold	3
Unseal Progress	1/3
Unseal Nonce	5f6ef075-ff8e-dacc-4bbf-ff51901ed729
Version	1.16.2
Build Date	2024-04-22T16:25:54Z
Storage Type	raft
HA Enabled	true

```
vault@vault-server-0:/$ vault operator unseal
```

```
Unseal Key (will be hidden):
```

Key	Value
---	-----
Seal Type	shamir
Initialized	true
Sealed	true
Total Shares	5
Threshold	3
Unseal Progress	2/3
Unseal Nonce	5f6ef075-ff8e-dacc-4bbf-ff51901ed729
Version	1.16.2
Build Date	2024-04-22T16:25:54Z
Storage Type	raft
HA Enabled	true

Либо можно распечатать Vault через UI интерфейс, для этого выполнить следующую команду:

```
kubectl -n vault port-forward vault-server-0 8200:8200
```

После чего UI интерфейс будет доступен по адресу <http://localhost:8200/>.

2.3.4 Расшифровка паролей

AOne использует SOPS для хранения и шифрования чувствительных данных.

Следует расшифровать пароли в директории `../bases/secrets` ключом с помощью команды:

```
sops -d -i secrets.yaml
```

2.4 Шаг 4. Установка и конфигурирование инфраструктурных компонентов

2.4.1 Подготовка окружения

Для подготовки окружения необходимо перейти в директорию `services`.

В репозитории хранится подготовленный файл `services/variables.tf`, который разворачивает в k8s кластере инфраструктуру, доступную по internal DNS с доменной зоной `*.adc.dns`.

Следует установить все переменные для кластера и хоста freeipa в зависимости от требований в `services/variables.tf`:

```
terraform init
terraform apply -target=module.get_info
```

После этого скопировать содержимое файла `infrastructure/etc_hosts` в системный файл `/etc/hosts`.

Содержимое файла `/etc/hosts` после редактирования:

```
##
# Host Database
#
# localhost is used to configure the loopback interface
# when the system is booting. Do not change this entry.
##
127.0.0.1        localhost
255.255.255.255 broadcasthost
::1             localhost
```

```
10.110.0.36 freeipa.freeipa.test
10.110.0.138 vault.adc.dns
10.110.0.138 keycloak.adc.dns
```

2.4.2 Конфигурирование Vault

Для конфигурирования Vault необходимо переименовать *services/terraform.tfvars.simple* в *services/terraform.tfvars* и заполнить поле token значением vault token6, полученным на шаге пп.2.3.3 . И выполнить команду:

```
terraform apply -target=module.vault
```

В результате сервис Vault, доступный по адресу *http://vault.adc.dns*, сконфигурируется и в него будут добавлены пароли, расшифрованные на шаге пп. 2.3.4 . В UI отображаются созданные Vault realms и vault kv.

2.4.3 Установка certificate-manager

Для хранения самоподписанных сертификатов проект использует *certificate-manager*, который необходимо устанавливать отдельно. Для его установки выполнить команду:

```
terraform apply -target=module.helm_prerequisites
```

Для проверки установки выполнить команду:

```
kubectl -n cert-manager get pods
```

Результат выполнения команды:

NAME	READY	STATUS	RESTARTS	AGE
cert-manager-75bf5d6774-j67n7	1/1	Running	0	79m
cert-manager-cainjector-6b9f7767b7-fqhvc	1/1	Running	0	79m
cert-manager-webhook-599d446d97-hjphm	1/1	Running	0	79m

2.4.4 Установка инфраструктурных компонентов

Проект AOne использует различные инфраструктурные компоненты, такие как postgres, keycloak, opensearch, vector, envoy, jaeger, temporal и другие.

Для установки инфраструктурных компонентов выполнить команду:

```
terraform apply -target=module.helm
```

Для проверки установки выполнить команду:

```
kubectl get pod --all-namespaces
```

Результат выполнения команды:

NAMESPACE	READY	STATUS	RESTARTS	NAME	AGE
ad-cloud	1/1	Running	1	postgresql-0	207d
api-gateway-dry-run-97x9d	1/1	Running	0	api-gateway-dry-run-deployment-7b84c6bb4c-97x9d	3h30m
api-gateway-dry-run-ghmwk	1/1	Running	0	api-gateway-dry-run-deployment-7b84c6bb4c-ghmwk	3h30m
api-gateway	1/1	Running	0	api-gateway-deployment-6b69bfdc7d-qw546	3h30m
api-gateway	1/1	Running	0	api-gateway-deployment-6b69bfdc7d-ww22v	3h30m
argocd	1/1	Running	1	argocd-application-controller-0	154d
argocd	1/1	Running	2 (13d ago)	argocd-applicationset-controller-6d65d44859-zqgmr	154d
argocd	1/1	Running	1	argocd-notifications-controller-6f89956f8f-dtwfw	154d
argocd	1/1	Running	1	argocd-redis-7648c74fb7-g9qjd	160d
argocd	1/1	Running	1	argocd-repo-server-7ffc8d4d98-q89qr	74d
argocd	1/1	Running	2 (13d ago)	argocd-server-66b89cf8d5-nr49f	154d
atlantis	1/1	Running	15 (13d ago)	atlantis-0	155d
audit	1/1	Running	16 (13d ago)	service-77cd56b5f5-xbnbf	33d
cerbos	1/1	Running	0	cerbos-6bbfb6f89b-cc58z	13d
cert-manager	1/1	Running	8	cert-manager-97fbb6ffd-r2q4l	183d
cert-manager	1/1	Running	28	cert-manager-cainjector-6b9f7767b7-mwblk	183d
cert-manager	1/1	Running	1	cert-manager-webhook-599d446d97-m8pft	183d
cloud-resource-manager-4ftxl	1/1	Running	0	cloud-resource-manager-deployment-6fcf5b94b9-4ftxl	12d
cloud-resource-manager-gfp59	1/1	Running	0	cloud-resource-manager-deployment-6fcf5b94b9-gfp59	12d
default-0-w4lnq	0/1	Completed	0	node-debugger-dev-adcloud-dev-adcloud-group-0-w4lnq	194d
default-0-wkbsj	0/1	Completed	0	node-debugger-dev-adcloud-dev-adcloud-group-0-wkbsj	194d
email	1/1	Running	12 (13d ago)	email-deployment-85db5d58bf-ssvtj	28d
envoy	1/1	Running	0	envoy-api-gateway-f5f9db7d7-8bw4s	13d
external-secrets	1/1	Running	2	external-secrets-847fbd65d7-pgmhm	47d
external-secrets				external-secrets-cert-controller-6c9d6c9798-	

zkts2		1/1	Running	2	154d
external-secrets				external-secrets-webhook-9fbd594d6-5tbnx	
1/1	Running	2		74d	
iam				iam-deployment-576dc6c9d8-gjjmv	
1/1	Running	0		43h	
ingress-nginx				ingress-nginx-controller-7ddd5cd8b9-2f76q	
1/1	Running	0		13d	
jaeger				jaeger-agent-4jwkp	
1/1	Running	1		35d	
jaeger				jaeger-agent-5kzcg	
1/1	Running	1		35d	
jaeger				jaeger-agent-6w2sr	
1/1	Running	1		49d	
jaeger				jaeger-agent-8hj7s	
1/1	Running	1		44d	
jaeger				jaeger-agent-fhvnf	
1/1	Running	1		68d	
jaeger				jaeger-agent-qqp6s	
1/1	Running	1		68d	
jaeger				jaeger-agent-tz2j8	
1/1	Running	2		68d	
jaeger				jaeger-agent-vmdbh	
1/1	Running	1		43d	
jaeger				jaeger-collector-74468d594f-59pcs	
1/1	Running	8 (13d ago)		47d	
jaeger				jaeger-ingester-688fb8cdf-f-kz6wf	
1/1	Running	20 (13d ago)		31d	
jaeger				jaeger-kafka-controller-0	
1/1	Running	1		68d	
jaeger				jaeger-kafka-controller-1	
1/1	Running	5 (13d ago)		47d	
jaeger				jaeger-kafka-controller-2	
1/1	Running	1		68d	
jaeger				jaeger-query-989686799-x5fpk	
2/2	Running	19 (13d ago)		68d	
keycloak-test				keycloak-test-0	
1/1	Running	15 (13d ago)		161d	
keycloak-test				keycloak-test-postgresql-0	
1/1	Running	1		166d	
keycloak				keycloak-0	
1/1	Running	12 (13d ago)		25d	
keycloak				keycloak-postgresql-0	
1/1	Running	1		207d	
kube-system				calico-kube-controllers-5cb9545b67-hfdkd	
1/1	Running	202		258d	
kube-system				calico-node-b2x28	
1/1	Running	1		43d	
kube-system				calico-node-c6csz	
1/1	Running	1		44d	
kube-system				calico-node-cr7cs	
1/1	Running	1		154d	
kube-system				calico-node-df8gp	
1/1	Running	2		215d	

kube-system			calico-node-kc9t7
1/1	Running	1	49d
kube-system			calico-node-lrfcr
1/1	Running	1	35d
kube-system			calico-node-rkrs8
1/1	Running	1	226d
kube-system			calico-node-sghkl
1/1	Running	1	35d
kube-system			calico-node-xh9v1
1/1	Running	249	252d
kube-system			calico-typha-55994d7c47-cpt5j
1/1	Running	13	258d
kube-system			cluster-autoscaler-769b74544-5spkt
1/1	Running	1288	258d
kube-system			coredns-c9hbl
1/1	Running	31	258d
kube-system			csi-cinder-controllerplugin-0
6/6	Running	22	258d
kube-system			csi-cinder-nodeplugin-88qf4
3/3	Running	3	154d
kube-system			csi-cinder-nodeplugin-crgrx
3/3	Running	3	49d
kube-system			csi-cinder-nodeplugin-jrzh5
3/3	Running	6	215d
kube-system			csi-cinder-nodeplugin-lj85f
3/3	Running	3	43d
kube-system			csi-cinder-nodeplugin-q9qm9
3/3	Running	3	35d
kube-system			csi-cinder-nodeplugin-rwv68
3/3	Running	3	35d
kube-system			csi-cinder-nodeplugin-sbr45
3/3	Running	9	258d
kube-system			csi-cinder-nodeplugin-t66tc
3/3	Running	3	44d
kube-system			csi-cinder-nodeplugin-zjw6n
3/3	Running	4	226d
kube-system			etcd-dev-adcloud-master-0
1/1	Running	3	258d
kube-system			etcd-events-dev-adcloud-master-0
1/1	Running	3	258d
kube-system			haproxy-dev-adcloud-dev-adcloud-group-0
1/1	Running	1	226d
kube-system			haproxy-dev-adcloud-dev-adcloud-group-1
1/1	Running	2	215d
kube-system			haproxy-dev-adcloud-dev-adcloud-group-2
1/1	Running	1	154d
kube-system			haproxy-dev-adcloud-dev-adcloud-group-3
1/1	Running	1	49d
kube-system			haproxy-dev-adcloud-dev-adcloud-group-4
1/1	Running	1	44d
kube-system			haproxy-dev-adcloud-dev-adcloud-group-5
1/1	Running	1	35d
kube-system			haproxy-dev-adcloud-dev-adcloud-group-6

1/1	Running	1	35d
kube-system			haproxy-dev-adcloud-dev-adcloud-group-7
1/1	Running	1	43d
kube-system			kube-apiserver-dev-adcloud-master-0
1/1	Running	3	258d
kube-system			kube-controller-manager-dev-adcloud-master-0
1/1	Running	1306	258d
kube-system			kube-proxy-44tfq
1/1	Running	3	258d
kube-system			kube-proxy-7dmt9
1/1	Running	2	226d
kube-system			kube-proxy-9hcpd
1/1	Running	1	35d
kube-system			kube-proxy-bgfjx
1/1	Running	1	154d
kube-system			kube-proxy-d7b2x
1/1	Running	1	49d
kube-system			kube-proxy-frmwq
1/1	Running	1	43d
kube-system			kube-proxy-lr5jt
1/1	Running	1	44d
kube-system			kube-proxy-pgq9b
1/1	Running	1	35d
kube-system			kube-proxy-zbf2l
1/1	Running	2	215d
kube-system			kube-scheduler-dev-adcloud-master-0
1/1	Running	1252	258d
kube-system			kubelet-csr-approver-6898c5b8c-bwltv
1/1	Running	1391	258d
kube-system			metrics-server-6f87f4f68c-t9nsc
1/1	Running	215	258d
kube-system			node-shell-0c63207f-148d-4b13-b7ba-
866c41122528	0/1	Completed	0 110d
kube-system			node-shell-0f2e27d5-78dc-4927-9043-
05cfbfdc1117	0/1	Completed	0 225d
kube-system			node-shell-0ff9728a-0cdb-4533-873c-
c26b08022dbf	0/1	Completed	0 71d
kube-system			node-shell-191e9d73-b4e4-4cc9-b5dc-
10dfb5577ad9	0/1	Completed	0 57d
kube-system			node-shell-30961893-64d7-4acb-a0ab-
9c02e58a83db	0/1	Completed	0 225d
kube-system			node-shell-3511b75f-7044-4f46-bb95-
d71da1dd73cb	0/1	Completed	0 158d
kube-system			node-shell-3fe968ef-d45d-4e15-a127-
1b7d72453415	0/1	Completed	0 189d
kube-system			node-shell-537af5b3-6467-4976-8b20-
db085785cd28	0/1	Completed	0 190d
kube-system			node-shell-5fb1d0c9-ddfd-422d-8f93-
32bdb6569e83	0/1	Completed	0 71d
kube-system			node-shell-637d19f9-7e32-41de-b9ba-
b12424256b62	0/1	Completed	0 240d
kube-system			node-shell-6f7b4886-a400-47e4-b7a6-
a3b45bc7c64d	0/1	Completed	0 71d

kube-system			node-shell-775d5c1f-5514-4087-9e74-
e5da0e10a439	0/1	Completed	0 158d
kube-system			node-shell-85aa9c78-c083-469a-b1f8-
af5e2c0b9a89	0/1	Completed	0 71d
kube-system			node-shell-8a5f7881-48ae-4a8b-9918-
15478763811d	0/1	Completed	0 71d
kube-system			node-shell-941107ff-0f11-4176-acb2-
e08a89289d19	0/1	Completed	0 188d
kube-system			node-shell-aea1828a-d8ae-417f-8254-
7658ad79dd13	0/1	Completed	0 110d
kube-system			node-shell-b37759f2-88c2-4ad2-87b8-
75ad198fcc0b	0/1	Completed	0 149d
kube-system			node-shell-b5ef76bd-2fe7-4ebd-8c60-
51190759f6af	0/1	Completed	0 88d
kube-system			node-shell-dc05b32b-f8fe-40c8-925c-
f26ede81ee81	0/1	Completed	0 71d
kube-system			node-shell-e23228d9-5869-4577-b3a7-
d78356f123f8	0/1	Completed	0 189d
kube-system			node-shell-ef3beb94-07c9-4093-9043-
047590ba7872	0/1	Completed	0 225d
kube-system			node-shell-f74edfbd-93ed-417c-bf23-
02c65ba437ab	0/1	Completed	0 145d
kube-system			openstack-cloud-controller-manager-ds2nz
1/1	Running	1279	258d
kube-system			shell-operator-7bb4b48449-8dp6h
1/1	Running	3	258d
kubernetes-dashboard			dashboard-metrics-scraper-7c8687b5d9-7d5x6
1/1	Running	3	258d
kubernetes-dashboard			kubernetes-dashboard-df7c84f45-dvklw
1/1	Running	117 (13d ago)	258d
lens-metrics			kube-state-metrics-847cd89659-dljmn
1/1	Running	3 (13d ago)	74d
lens-metrics			node-exporter-29qh8
1/1	Running	1	49d
lens-metrics			node-exporter-4ngbb
1/1	Running	2	127d
lens-metrics			node-exporter-7kfxk
1/1	Running	1	35d
lens-metrics			node-exporter-ddskt
1/1	Running	1	35d
lens-metrics			node-exporter-h4n5m
1/1	Running	1	127d
lens-metrics			node-exporter-kcfxc
1/1	Running	1	44d
lens-metrics			node-exporter-mh4kp
1/1	Running	2	127d
lens-metrics			node-exporter-pmrp6
1/1	Running	2	127d
lens-metrics			node-exporter-q9qnw
1/1	Running	1	43d
lens-metrics			prometheus-0
1/1	Running	1	47d
logging			cluster-46xx2

1/1	Running	16 (13d ago)	43d	
logging			cluster-fxlxz	
1/1	Running	16 (13d ago)	44d	
logging			cluster-kgl2v	
1/1	Running	15 (13d ago)	68d	
logging			cluster-ldltx	
1/1	Running	16 (13d ago)	49d	
logging			cluster-nwhmp	
1/1	Running	16 (13d ago)	35d	
logging			cluster-pq4cj	
1/1	Running	15 (13d ago)	68d	
logging			cluster-v6rsr	
1/1	Running	16 (13d ago)	68d	
logging			cluster-wvfh2	
1/1	Running	15 (13d ago)	35d	
monitoring			alertmanager-0	
1/1	Running	1	47d	
monitoring			grafana-95bf9d57b-g76b7	
1/1	Running	2 (12d ago)	47d	
monitoring			prometheus-node-exporter-658sp	
1/1	Running	1	43d	
monitoring			prometheus-node-exporter-bvw5k	
1/1	Running	1	90d	
monitoring			prometheus-node-exporter-d9cvh	
1/1	Running	1	44d	
monitoring			prometheus-node-exporter-gr7df	
1/1	Running	1	90d	
monitoring			prometheus-node-exporter-lwv7t	
1/1	Running	1	90d	
monitoring			prometheus-node-exporter-m9g6q	
1/1	Running	1	35d	
monitoring			prometheus-node-exporter-p2t4f	
1/1	Running	2	90d	
monitoring			prometheus-node-exporter-v5m49	
1/1	Running	1	35d	
monitoring			prometheus-node-exporter-xh7mp	
1/1	Running	1	49d	
monitoring			vmagent-victoria-metrics-agent-b79bfb4df-	
5tpqt		0/1	Completed	0 35d
monitoring			vmagent-victoria-metrics-agent-b79bfb4df-	
d2tbc		0/1	Completed	0 74d
monitoring			vmagent-victoria-metrics-agent-b79bfb4df-	
n8fxh		0/1	Completed	0 32d
monitoring			vmagent-victoria-metrics-agent-b79bfb4df-	
t2qvw		1/1	Running	1 29d
monitoring			vmagent-victoria-metrics-agent-b79bfb4df-	
tg689		0/1	Completed	0 48d
monitoring			vmagent-victoria-metrics-agent-b79bfb4df-	
w2vbg		0/1	Completed	0 44d
monitoring			vmalert-victoria-metrics-alert-server-	
86dc8ff464-jchb9		1/1	Running	1 47d
monitoring			vmauth-victoria-metrics-auth-5b69fd56ff-vn5rh	
1/1	Running	1	47d	

monitoring			vmcluster-victoria-metrics-cluster-vminsert-
86484d4464-5w19p	1/1	Running	1 28d
monitoring			vmcluster-victoria-metrics-cluster-vminsert-
86484d4464-pvx94	1/1	Running	1 28d
monitoring			vmcluster-victoria-metrics-cluster-vmselect-
5ff947566d-2g7h9	1/1	Running	1 28d
monitoring			vmcluster-victoria-metrics-cluster-vmselect-
5ff947566d-7twzf	1/1	Running	1 28d
monitoring			vmcluster-victoria-metrics-cluster-vmstorage-
0	1/1	Running	1 28d
nexus			nexus-nexus-repository-manager-b8c97cb45-
lqrb7	1/1	Running	1 29d
opa-gatekeeper			gatekeeper-audit-77dc4cbfdf-j9xfl
1/1	Running	200	258d
opa-gatekeeper			gatekeeper-controller-manager-746bbf6679-
65rz8	1/1	Running	199 258d
openfga			openfga-7d4b7d4bdb-4jvlj
1/1	Running	31 (13d ago)	46d
openfga			openfga-7d4b7d4bdb-m792d
1/1	Running	26 (13d ago)	31d
openfga			openfga-7d4b7d4bdb-tldfm
1/1	Running	35 (13d ago)	68d
opensearch			dashboard-7786999c98-4lgvm
1/1	Running	1	47d
opensearch			master-0
1/1	Running	1	47d
opensearch			master-1
1/1	Running	1	68d
opensearch			master-2
1/1	Running	1	29d
postgresql			ad-pgpool-7cc9cf97f8-s24xj
1/1	Running	1	27d
postgresql			ad-postgresql-0
1/1	Running	1	27d
product-resource-manager			product-resource-manager-deployment-
59dd97d4f5-6tcr4	1/1	Running	0 13m
resource-manager-api-dry-run			resource-manager-api-dry-run-deployment-
554b585c44-4v55z	1/1	Running	0 3h31m
resource-manager-api			resource-manager-api-deployment-66c5949f84-
9chzw	1/1	Running	0 3h30m
resource-manager			resource-manager-deployment-cc67d754c-rdv5j
1/1	Running	0	3h30m
temporal			temporal-admintools-778fdfffc-8qhqq
1/1	Running	2	74d
temporal			temporal-frontend-5d9745d875-jbg6w
1/1	Running	7 (13d ago)	27d
temporal			temporal-grafana-777dcdd975-7hm7q
1/1	Running	1	27d
temporal			temporal-history-978b8d84c-n5xqw
1/1	Running	7 (13d ago)	27d
temporal			temporal-matching-55fbd68c65-bsv8w
1/1	Running	7 (13d ago)	27d
temporal			temporal-prometheus-node-exporter-2dm8l

1/1	Running	1	154d
temporal			temporal-prometheus-node-exporter-5x647
1/1	Running	1	43d
temporal			temporal-prometheus-node-exporter-6d74n
1/1	Running	2	216d
temporal			temporal-prometheus-node-exporter-6nxcf
1/1	Running	1	44d
temporal			temporal-prometheus-node-exporter-bs8x2
1/1	Running	1	35d
temporal			temporal-prometheus-node-exporter-fflv9
1/1	Running	2	215d
temporal			temporal-prometheus-node-exporter-jhq4
1/1	Running	1	35d
temporal			temporal-prometheus-node-exporter-kqtv9
1/1	Running	1	49d
temporal			temporal-prometheus-server-7669b476d6-bx92h
2/2	Running	2	74d
temporal			temporal-web-7f8b77f6bf-m6pbn
1/1	Running	1	33d
temporal			temporal-worker-799d7b67f6-2wm65
1/1	Running	7 (13d ago)	27d
ui-core-app-auth			ui-core-app-auth-deployment-7f567db5c-l4m4w
1/1	Running	0	3h31m
ui-core-app-auth			ui-core-app-auth-deployment-7f567db5c-lfkfr
1/1	Running	0	3h30m
vault			vault-injector-85649787cb-6rj77
1/1	Running	2	74d
vault			vault-secrets-webhook-57ddcd558-mgl88
1/1	Running	1	28d
vault			vault-server-0
1/1	Running	1	165d

2.4.5 Установка дополнительных инфраструктурных компонентов

Проект AOne использует дополнительные инфраструктурные компоненты с циклическими зависимостями, которые необходимо устанавливать отдельно.

Для установки инфраструктурных компонентов выполнить команду:

```
terraform apply -target=module.helm_postinstall
```

2.5 Шаг 5. Установка приложений

Проект AOne использует различные приложения, такие как `api_gateway`, `resource_manager`, `cloud_resource_manager`, `iam` другие. Для установки приложений выполнить команду:

```
terraform apply -target=module.products
```