

- Автоматически происходит влитие текущего состояния ветки master в ветку release
- Фиксы
 - Создаётся запрос на принятие изменений в ветку release
 - Прохождение проверок для запроса на принятие изменений (PR)
 - Влитие изменений в ветку testing
 - Автоматическое создание запроса на влитие фикса в ветку master
 - В случае успешного прохождения проверок (без approve) происходит автоматическое влитие в master
 - В случае
- Список основных контуров развёртывания
 - Контур разработки
Контур для разработки микросервисов. Предоставляются по запросу разработчиков
 - CI/CD контуры
Контур на которых осуществляется автоматическое тестирование микросервисов службами CI/CD
 - Тестовый контур
Контур на котором осуществляется тестирование новых фич в ручном или полуавтоматическом режиме
 - Pre-production контур
Контур который является копией состояния production контура и на котором проводится тестирование в ручном или полуавтоматическом режиме, а также профилирование
 - Production контур
Контур для доступа конечных пользователей
-

Требования к конфигурации микросервисов

- Настройки микросервисов хранятся централизованно (кроме задач изолированного тестирования)
- Настройки микросервисов могут изменяться динамически
- В качестве сервера конфигурации используется кластер Etcd (далее по тексту Etcd Configuration Cluster)
- Параметры конфигурации хранятся на сервере конфигурации в следующем виде:
<название_пространства_ключей>.<имя микросервиса>.
[имя_группы.]<имя_параметра>. Имена микросервисов и переменных должны быть

представлены в стиле snake_case (нижний регистр, вместо пробела нижнее подчеркивание). Например, service.service_cache.cache_size=256G

- Для каждого микросервиса ОБЯЗАН присутствовать параметр "название_пространства_ключей.имя_микросервиса.declaration" в виде json схемы, где перечисляются параметры конфигурации, их описание и представление в HTML форме (см. <https://jsonform.github.io/jsonform/playground/index.html>). В дополнение к стандартным свойствам JSON схемы, каждый параметр должен иметь атрибут "access" тип "string", который определяет кто может редактировать данный параметр:
 - internal — может модифицировать только сам микросервис
 - superuser — только администраторы платформы
 - user — пользователи платформы

Пример:

microservice1.declaration

```
{
  "schema": {
    "название_пространства_ключей.microservice1.field1": {
      "type": "string",
      "title": "A field 1",
      "access": "superuser"
    },
    "название_пространства_ключей.microservice1.field2": {
      "type": "integer",
      "title": "A field 2",
      "access": "superuser"
    }
  },
  "form": [
    {
      "key": "название_пространства_ключей.microservice1.field1"
    },
    {
      "key": "название_пространства_ключей.microservice1.field2",
    },
    {
      "type": "submit",
      "title": "Submit"
    }
  ]
}
```

- Микросервис может получать информацию о конфигурации из файла или от централизованного сервера конфигурации etcd
 - В случае использования сервера конфигурации микросервис при запуске ОБЯЗАН получить в качестве аргумента командной строки параметр --configuration-cluster-endpoints=<список>, в котором в виде списка задаются пары хост:порт. Разделителем является запятая. Пример: --configuration-cluster-endpoints="aegees-config-1.local:6379, aegees-config-2.local:6379".
 - Для задач тестирования предусмотрена возможность получения конфигурации из файла. В этом случае микросервис ОБЯЗАН получить в качестве аргумента командной строки параметр --configuration-file=<путь к файлу>. Формат файла JSON.
 - В случае, если ни один из вышеперечисленных параметров не задан, микросервис должен вывести в stderr сообщение об ошибке и завершиться.
- Для доступа к серверу аутентификации используется

- При запуске микросервис ОБЯЗАН получить в качестве аргумента командной строки параметр `--certificate=<путь_к_сертификату>`. Common name из сертификата будет использоваться в качестве keystore сервером конфигурации. В случае, если параметр не задан, микросервис должен вывести в stderr сообщение об ошибке и завершиться.
- При запуске микросервис ОБЯЗАН получить в качестве аргумента командной строки параметр `--privatekey=<путь_к_приватному_ключу>`. В случае, если параметр не задан, микросервис должен вывести в stderr сообщение об ошибке и завершиться.
- При запуске микросервис ОБЯЗАН получить в качестве аргумента командной строки параметр `--ca-certificate=<путь_к_сертификату>`. В случае, если параметр не задан, микросервис должен вывести в stderr сообщение об ошибке и завершиться.
- В том случае если микросервис шардируемый, микросервис ОБЯЗАН получить в качестве аргумента командной строки параметр `--shard-number=<номер>`, где номер начинается с 1. В случае, если параметр не задан, микросервис должен вывести в stderr сообщение об ошибке и завершиться.
- При старте микросервис ОБЯЗАН выполнить следующие действия:
 - подписаться на канал
"название_пространства_ключей.notification.имя_микросервиса.CONFIGURATION_CHANGED"
 - подписаться на канал
"название_пространства_ключей.notification.имя_микросервиса.SUSPEND"
 - подписаться на канал
"название_пространства_ключей.notification.имя_микросервиса.RESTART"
 - подписаться на канал
"название_пространства_ключей.notification.имя_микросервиса.SHUTDOWN"
 - подписаться на другие нотификации, если это необходимо, например, если сервис использует параметры других сервисов
 - запросить список параметров
"название_пространства_ключей.имя_микросервиса.*" и параметры других сервисов, если необходимо
 - запросить значение
"название_пространства_ключей.notification.имя_микросервиса.SUSPEND"
 - проинициализировать своё состояние
 - в зависимости от значения
"название_пространства_ключей.notification.имя_микросервиса.SUSPEND" перейти к штатной работе или ожидать изменения этого параметра