

Программное обеспечение «Тиражная служба «Специгра»

Руководство для установки и эксплуатации программного обеспечения

Содержание

| | |
|--|----|
| 1. Инфраструктура для Тиражной службы «Специгра» | 3 |
| 2. Описание модулей Тиражной службы «Специгра» | 3 |
| 3. Установка Тиражной службы «Специгра» | 9 |
| 4. Выключение/включение модулей | 10 |

1. Инфраструктура для Тиражной службы «Специгра»

1. Хост с сервисом MariaDB (для хранения данных службы) версия MariaDB 10.4.14
2. Хост с сервисом ActiveMQ
3. Хост с сервисом SSH (для обмена с внутренними пользователями/системами)
4. Система оркестрации Kubernetes (для запуска модулей службы) версия Kubernetes v1.20.15 версия Ingress <https://github.com/kubernetes/ingress-nginx/releases/tag/ingress-nginx-2.11.1>
5. Сервис DNS (для резолвинга имён разных сервисов, используемых в службе)

2. Описание модулей Тиражной службы «Специгра»

Модуль vps-migration

Модуль системы отвечающий за создание и обновление БД.

Параметры подключения к БД передаются через переменные среды

Модуль vps-srv

Главный модуль системы.

Конфигурация:

```
{
  "ConnectionStrings": {
    "FixPriceConnectionString": "server=vps-
db.website.ru;userid=vps_user;pwd=vps_user;database=vps_db;
sslmode=none;charset=utf8;allowZeroDateTime=false;convert zero
datetime=True;AllowPublicKeyRetrieval=True;Allow
User Variables=true;old guides=true; Max Pool Size=25;Connection
Timeout=10;UseAffectedRows=True;"
  },
  "GateWinResultSftpConfig": {
    "Username": "user",
    "Password": "",
    "Host": "ssh.website.ru",
    "Port": 22,
    "KeyFile": "./rsa/rsakey"
  },
  "ClientConfig": {
    "Mask": "http://{{ .Values.env }}-vps-{svc}-svc.{{ .Values.env }}-
vps.svc.cluster.local"
  },
  "BrokerHostConfig": {
    "Login": "draw_service",
    "Password": "",

```

```
"Url":
"activemq:failover:(tcp://broker1.website.ru:61616,tcp://broker2.website.ru)"
,
"Queue": "queue://TRN.SALES::DRAW_SERVICE"
},
"QuartzConfig": {
"quartz.scheduler.instanceId": "AUTO",
"quartz.scheduler.instanceName": "fixprice",
"quartz.serializer.type": "json",
"quartz.jobStore.dataSource": "myDS",
"quartz.jobStore.type": "Quartz.Impl.AdoJobStore.JobStoreTX, Quartz",
"quartz.jobStore.driverDelegateType": "Quartz.Impl.AdoJobStore.MySQLDelegate,
Quartz",
"quartz.jobStore.tablePrefix": "QRTZ_",
"quartz.jobStore.misfireThreshold": "1000",
"quartz.jobStore.clustered": "true",
"quartz.jobStore.clusterCheckinInterval": "1000",
"quartz.dataSource.myDS.connectionProvider.type":
"Stoloto.Dice.Scheduler.Implementation.MySqlDbProvider,
Stoloto.Dice.Scheduler"
},
"FixPriceConfig": {
"Products": [ 11111 ],
"StartDelay": "00:00:15",
"GenerateTicketStatusInterval": "00:00:05",
"UploadTicketStatusInterval": "00:00:05",
"ConfirmTicketsInterval": "00:00:15",
"RequestCombinationInterval": "00:00:05",
"WinUploadConfirmInterval": "00:00:15",
"CalculateResultTimeout": "00:15:00",
"GenerateTicketTimeout": "00:15:00",
"UploadTicketTimeout": "00:15:00",
"TempDirectory": "./",
"WinnerRemoteDirectory": "/home/user"
},
"Serilog": {
"MinimumLevel": {
"Default": "Information",
"Override": {
"Stoloto": "Verbose",
"Microsoft": "Information",
"Microsoft.AspNetCore": "Error",
"Quartz": "Information",
"System.Net.Http.HttpClient": "Information",
"Stoloto.FixPrice.Queries.FixPriceGetTemplateHandler": "Information",
"Stoloto.FixPrice.Queries.FixPriceGetTemplatesHandler": "Information",
"Stoloto.FixPrice.Queries.FixPriceGetDrawsHandler": "Information"
}
}
},
"WriteTo": [
{
"Name": "Console",
"Args": {
"outputTemplate": "[{Level:u3}] {Timestamp:MM-dd HH:mm:ss} {TraceId}
{SourceContext:l} {Message:l}
{NewLine}{Exception}"
}
}
]
},
```

```
"HttpServicesConfig": {
  "Services": [
    {
      "Srv": "srv",
      "Address": "http://{{ .Values.env }}-vps-srv-svc.{{ .Values.env }}-vps.svc.cluster.local"
    },
    {
      "Srv": "tgen",
      "Address": "http://{{ .Values.env }}-vps-tgen-svc.{{ .Values.env }}-vps.svc.cluster.local"
    },
    {
      "Srv": "rng",
      "Address": "https://vps-rng-service.{{ .Values.env }}-vps.svc.cluster.local"
    }
  ],
  "OnError": {
    "RetryCount": 3,
    "Timeout": "00:00:01",
    "Jitter": "00:00:01"
  }
}
```

Модуль vps-tgen

Модуль занимается генерацией билетов.

Конфигурация:

```
{
  "ConnectionStrings": {
    "TicketConnectionString": "server=vps-db.website.ru;userid=vps_tgen_user;pwd=vps_tgen_user;database=vps_db;sslmode=none;charset=utf8;allowZeroDateTime=false;convert zero datetime=True;AllowPublicKeyRetrieval=True;Allow User Variables=true;old guides=true; Max Pool Size=25;Connection Timeout=10;UseAffectedRows=True;"
  },
  "TicketsConfig": {
    "TempDirectory": "./",
    "TicketRemoteDirectory": "/home/user"
  },
  "GateTicketSftpConfig": {
    "Username": "user",
    "Password": "",
    "Host": "{{ .Values.env }}-gate-rsync.{{ .Values.env }}.gate.orglot.office",
    "Port": 22,
    "KeyFile": "./rsa/rsakey"
  },
  "Serilog": {
    "MinimumLevel": {
      "Default": "Information",
      "Override": {
        "Stoloto": "Verbose",
        "Microsoft": "Information",
        "Microsoft.AspNetCore": "Error",
        "Quartz": "Information",

```

```
"System.Net.Http.HttpClient": "Information",
}
},
"WriteTo": [
{
"Name": "Console",
"Args": {
"outputTemplate": "[{Level:u3}] {Timestamp:MM-dd HH:mm:ss} {TraceId}
{SourceContext:l} {Message:l}j
{NewLine}{Exception}"
}
}
]
}
}
```

Модуль vps-events

Конфигурация:

```
{
"ConnectionStrings": {
"WebEventsConnectionString": "server=vps-
db.website.ru;userid=vps_events_user;pwd=vps_events_user;
database=vps_db;sslmode=none;charset=utf8;allowZeroDateTime=false;convert
zero datetime=True;
AllowPublicKeyRetrieval=True;Allow User Variables=true;old guides=true; Max
Pool Size=25;Connection Timeout=10;
UseAffectedRows=True;"
},
"WebEventPollConfig": {
"SendEvents": true,
"SendBalance": true
},
"SendEventsConfig": {
"BaseAddress": "http://ifd.website.ru",
"Path": "tasktype_endpoint/send_draw_events_and_draw_results"
},
"SendBalanceConfig": {
"BaseAddress": "http://ifd.website.ru",
"Path": "tasktype_endpoint/send_draw_events_and_draw_results",
"Timeout": "00:00:05"
},
"WwwSftpClientConfig": {
"Username": "user",
"Host": "ssh.website.ru",
"Port": 22,
"KeyFile": "./rsa/rsakey"
},
"WebEventConfig": {
"WinUploadRemoteDirectory": "/home/user",
"TempDirectory": "/"
},
"Serilog": {
"MinimumLevel": {
"Default": "Information",
"Override": {
"Stoloto": "Verbose",
"Microsoft": "Information",
```

```
"Microsoft.AspNetCore": "Error",
"System.Net.Http.HttpClient": "Information"
},
"WriteTo": [
{
"Name": "Console",
"Args": {
"outputTemplate": "[{Level:u3}] {Timestamp:MM-dd HH:mm:ss} {TraceId}
{SourceContext:l} {Message:l}
{NewLine}{Exception}"
}
}
]
```

Модуль vps-archive

Конфигурация:

```
{
"ConnectionStrings": {
"ArchiveConnectionString": "vps-
db.website.ru;userid=vps_archive_user;pwd=vps_archive_user;database=vps_db;
sslmode=none;charset=utf8;allowZeroDateTime=false;convert zero
datetime=True;AllowPublicKeyRetrieval=True;Allow
User Variables=true;old guides=true; Max Pool Size=25;Connection
Timeout=10;UseAffectedRows=True;"
},
"ArchiveConfig": {
"WinnerUploadBatchSize": 5000
},
"QuartzConfig": {
"quartz.threadPool.threadCount": 1
},
"Serilog": {
"MinimumLevel": {
"Default": "Information",
"Override": {
"Stoloto": "Verbose",
"Microsoft": "Information",
"Microsoft.AspNetCore": "Error",
"Quartz": "Information",
"System.Net.Http.HttpClient": "Information"
}
}
},
"WriteTo": [
{
"Name": "Console",
"Args": {
"outputTemplate": "[{Level:u3}] {Timestamp:MM-dd HH:mm:ss} {TraceId}
{SourceContext:l} {Message:l}
{NewLine}{Exception}"
}
}
]
```

Модуль vps-mg

Конфигурация:

```
{
  "ConnectionStrings": {
    "RngConnectionString": "server=vps-
db.website.ru;userid=vps_rng_user;pwd=vps_rng_user;database=vps_db;
sslmode=none;charset=utf8;allowZeroDateTime=false;convert zero
datetime=True;AllowPublicKeyRetrieval=True;Allow
User Variables=true;old guides=true; Max Pool Size=25;Connection
Timeout=10;UseAffectedRows=True;"
  },
  "AccordRngConfig": {
    "Primary": {
      "HostAddress": "rng1.website.ru",
      "Port": 6
    },
    "Fallback": {
      "HostAddress": "rng2.website.ru",
      "Port": 6
    }
  },
  "Serilog": {
    "MinimumLevel": {
      "Default": "Information",
      "Override": {
        "Stoloto": "Verbose",
        "Microsoft": "Information",
        "Microsoft.AspNetCore": "Error",
        "Quartz": "Information",
        "System.Net.Http.HttpClient": "Information"
      }
    }
  },
  "WriteTo": [
    {
      "Name": "Console",
      "Args": {
        "outputTemplate": "[{Level:u3}] {Timestamp:MM-dd HH:mm:ss} {TraceId}
{SourceContext:l} {Message:l}
{NewLine}{Exception}"
      }
    }
  ]
}
```

Модуль vps-client

Модуль обеспечивающий интерфейс между оператором системы и модулем vps-srv через web интерфейс

```
{
  "production": true,
  "urlSrv": "http://vps.website.ru/api",
  "urlRng": "http://vps.website.ru/api",
  "showDevActions": false,
```



```
"showTestActions": false
}
```

Ingress

Сущность в Kubernetes обеспечивающая маршрутизацию внешнего HTTP и HTTPS трафика кластера Kubernetes среди модулей vps-srv и vpsclient внутри кластера

3. Установка Тиражной службы «Специгра»

1. Перенос артефактов в локальный registry

```
$ sudo docker image load --input vps-migration.tar.gz
$ sudo docker image load --input vps-srv.tar.gz
$ sudo docker image load --input vps-tgen.tar.gz
$ sudo docker image load --input vps-events.tar.gz
$ sudo docker image load --input vps-archive.tar.gz
$ sudo docker image load --input vps-client.tar.gz
$ sudo docker image tag original-registry.example.org/vps-migration:0.1.0-dev.1152 new-registry.example.com/vps-migration:0.1.0-dev.1152
$ sudo docker image tag original-registry.example.org/vps-srv:0.1.0-dev.1152 new-registry.example.com/vps-srv:0.1.0-dev.1152
$ sudo docker image tag original-registry.example.org/vps-tgen:0.1.0-dev.1152 new-registry.example.com/vps-tgen:0.1.0-dev.1152
$ sudo docker image tag original-registry.example.org/vps-events:0.1.0-dev.1152 new-registry.example.com/vps-events:0.1.0-dev.1152
$ sudo docker image tag original-registry.example.org/vps-archive:0.1.0-dev.1152 new-registry.example.com/vps-archive:0.1.0-dev.1152
$ sudo docker image tag original-registry.example.org/vps-client:0.1.0-dev.1152 new-registry.example.com/vps-client:0.1.0-dev.1152
$ sudo docker push new-registry.example.com/vps-migration
$ sudo docker push new-registry.example.com/vps-srv
$ sudo docker push new-registry.example.com/vps-tgen
$ sudo docker push new-registry.example.com/vps-events
$ sudo docker push new-registry.example.com/vps-archive
$ sudo docker push new-registry.example.com/vps-client
```

2. Создание namespace

```
$ kubectl create namespace vps
```

3. Создание секрета для доступа к registry

```
$ kubectl create secret generic regcred --from-file=.dockerconfigjson=/tmp/config.json --type=kubernetes.io/dockerconfigjson --kubeconfig ~/.kube/kubeconfig -namespace vps
$ cat /tmp/config.json
{"auths":{"new-registry.example.com":{"username":"registry_user","password":"password","authcmVnaXN0cnlfdXNlcjpwYXNzd29yZAo="}}}
```

4. Установка модулей

a. Установка vps-migration

b. Установка vps-srv

```
$ helm upgrade --install vps-srv vps-srv -f vps-srv/values.yaml --kubeconfig  
~/.kube/kubeconfig --  
namespace vps
```

c. Установка vps-tgen

```
$ helm upgrade --install vps-tgen vps-tgen -f vps-tgen/values.yaml --  
kubeconfig ~/.kube/kubeconfig  
--namespace vps
```

d. Установка vps-events

```
$ helm upgrade --install vps-events vps-events -f vps-events/values.yaml --  
kubeconfig ~/.kube  
/kubeconfig --namespace vps
```

e. Установка vps-archive

```
$ helm upgrade --install vps-archive vps-archive -f vps-archive/values.yaml -  
-kubeconfig ~/.kube  
/kubeconfig --namespace vps
```

f. Установка vps-client

```
$ helm upgrade --install vps-client vps-client -f vps-client/values.yaml --  
kubeconfig ~/.kube  
/kubeconfig --namespace vps
```

g. Установка ingress

```
$ helm upgrade --install vps-ingress vps-ingress -f vps-ingress/values.yaml -  
-kubeconfig ~/.kube  
/kubeconfig --namespace vps
```

h. Добавление ДНС записи для ip адреса ingress

```
vps.website.ru. IN A 1.2.3.4
```

4. Выключение/включение модулей

На примере vps-srv

```
#  
kubectl scale --replicas=0 deployment/vps-srv --kubeconfig ~/.kube/kubeconfig  
--namespace vps  
#  
kubectl scale --replicas=1 deployment/vps-srv --kubeconfig ~/.kube/kubeconfig  
--namespace vps
```