

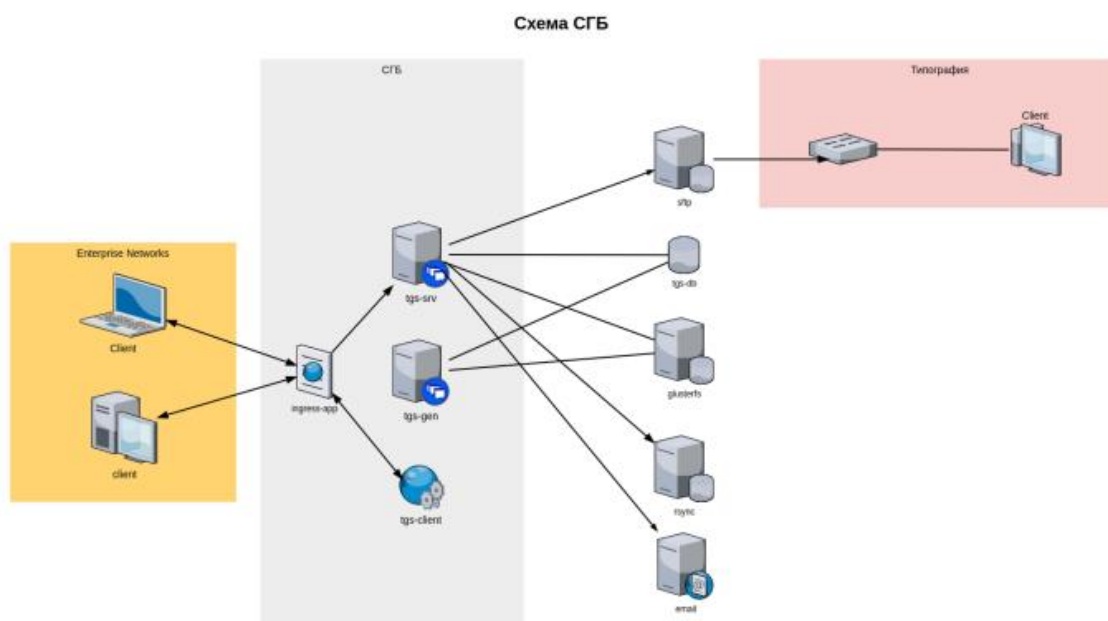
Программное обеспечение «Служба генерации билетов»

Руководство для установки и эксплуатации программного обеспечения

Оглавление

1. Инфраструктура для СГБ.....	3
2. Описание модулей службы генерации билетов.....	3
3. Установка СГБ.....	6

1. Инфраструктура для СГБ



1. Хост с сервисом Postresql (для хранения данных службы) версия PostgreSQL 12.8
2. Хост с сервисом SFTP (для обмена с внешними пользователями/системами)
3. Хост с сервисом SSH (для обмена с внутренними пользователями/системами)
4. Хост с сервисом электронной почты (для отправки писем внешним пользователям)
5. Разделяемая файловая система (для обмена файлами, tgs-gen генерирует файлы, tgs-srv их обрабатывает)
6. 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>
7. Сервис DNS (для резолвинга имён разных сервисов используемых в службе)

2. Описание модулей службы генерации билетов

Модуль tgs-srv

Главный модуль системы. Обслуживает все поступающие запросы, выгружает сгенерированные билеты в виде файлов и отправляет уведомления по электронной почте.

Конфигурация:

```
{
"ConnectionStrings": {
```

```
"AppConnectionString": "User ID=tgs_user;Password=tgs_user_password;Host=tgs-  
db.website.ru;Port=5432;  
Database=tgs_db;Pooling=true;",  
"MigrationConnectionString": "User ID=admin;Password=admin;Host=tgs-  
db.website.ru;Port=5432;Database=tgs_db;  
Pooling=true;",  
"BusConnectionString": "User ID=bus_user;Password=bus_user_password;Host=tgs-  
db.website.ru;Port=5432;  
Database=tgs_db;Pooling=true;"  
},  
"ApiConfig": {  
"CorsAllowOrigin": "http://tgs.website.ru",  
"GenBasePublicUrl": "http://tgs.website.ru",  
"ApiBasePublicUrl": "http://tgs.website.ru",  
"GenBaseUrl": "http://kubernetes-tgs-gen-svc.kubernetes-tgs.svc.cluster.local",  
"BaseAttachmentPath": "/tmp/gluster/kubernetes",  
"BasePath": "/tmp/gluster/kubernetes",  
"FireDomainEventImmediately": true,  
"ApplyMigrationsOnRestart": true,  
"MockGateUploadConfirm": false,  
"StartGateUploadConfirmService": true,  
"ReplicateBuEvents": true,  
"UseChannelQueue": false,  
"SendEmail": true,  
"RemoteGateDirectory": "/rsync/",  
"UseLocalGateDirectory": false,  
"IsNeedCheckConnections": true,  
"IsNeedArchiveTypographyFilesByCopy": true,  
"SequenceOptions": {  
"Orders": {  
"PrimaryKey": "Id",  
"StartValue": "10142200"  
}  
},  
"Email": {  
"DownloadTicketsLink": "http://tgs.website.ru",  
"TicketControlServiceEmail": "",  
"ApplicationEmail": "tgs@email.ru",  
"ApplicationUserName": "tgs",  
"ApplicationEmailNickName": " ",  
"ApplicationEmailPassword": "tgspassword",  
"SmtpServerAddress": "email.ru",  
"SmtpServerPort": "25"  
}  
},  
"TypographySftpConfig": {  
"Username": "sftp",  
"Host": "tgs-sftp.website.ru",  
"Port": 2222,
```

```
"KeyFile": "./rsa/sftp-rsakey"
},
"GateSftpConfig": {
"Username": "rsync",
"Host": "tgs-rsync.website.ru",
"Port": 22,
"Password": "dumb",
"KeyFile": "./rsa/rsync-rsakey"
},
"GateScpConfig": {
"Username": "rsync",
"Host": "tgs-rsync.website.ru",
"Port": 22,
"Password": "dumb",
"KeyFile": "./rsa/rsync-rsakey"
}
}
```

Модуль tgs-gen

Модуль занимается исключительно генерацией билетов.

Конфигурация

```
{
"GenConfig": {
"ApiBaseUrl": "http://kubernetes-tgs-srv-svc.kubernetes-tgs.svc.cluster.local",
"BasePath": "/tmp/gluster/kubernetes",
"CopyGateFiles": false,
"CopyTypographyFiles": false,
"RemoteGateDirectory": "",
"RemoteTypographyDirectory": ""
}
}
```

Модуль tgs-client

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

Конфигурация:

```
{
"production": true,
"api": "https://tgs.website.ru/api",
```

```
"printHouseFeatures": true,  
"enableDeveloperMode": false,  
"newYearVersion": "auto"  
}
```

Ingress

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

3. Установка СГБ

1. Перенос артефактов в локальный registry

```
$ sudo docker image load --input tgs-srv.tar.gz  
$ sudo docker image load --input tgs-gen.tar.gz  
$ sudo docker image load --input tgs-client.tar.gz  
$ sudo docker image tag original-registry.example.org/tgs-srv:0.1.0-dev.1152 new-  
registry.example.com  
/tgs-srv:0.1.0-dev.1152  
$ sudo docker image tag original-registry.example.org/tgs-gen:0.1.0-dev.1152 new-  
registry.example.com  
/tgs-gen:0.1.0-dev.1152  
$ sudo docker image tag original-registry.example.org/tgs-client:0.1.0-dev.1152 new-  
registry.example.com  
/tgs-client:0.1.0-dev.1152  
$ sudo docker push new-registry.example.com/tgs-srv  
$ sudo docker push new-registry.example.com/tgs-gen  
$ sudo docker push new-registry.example.com/tgs-client
```

2. Создание namespace

```
$ kubectl create namespace tgs
```

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 tgs  
$ cat /tmp/config.json  
{"auths":{"new-registry.example.com":{"username":"registry_user","password":"password","auth":"  
cmVnaXN0cnlfdXNlcjpwYXNzd29yZAo="}}}
```

4. Установка модулей

а. Установка tgs-srv

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

б. Установка tgs-gen

```
$ helm upgrade --install tgs-gen tgs-gen -f tgs-gen/values.yaml --kubeconfig ~/.kube/kubeconfig --namespace tgs
```

в. Установка tgs-client

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

г. Установка ingress

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

е. Добавление DNS записи для ip адреса ingress

```
tgs.website.ru. IN A 1.2.3.4
```