

ПРОТОКОЛ № 21260/2024

проведения совместных испытаний программного обеспечения «AlphaBI» версии 5.3.0 и операционной системы специального назначения «Astra Linux Special Edition» РУСБ.10015-01 (очередное обновление 1.8)

г. Санкт-Петербург

21.05.2024

1 Предмет испытаний

1.1 В настоящем протоколе зафиксирован факт проведения в период с 17.04.2024 по 17.05.2024 совместных испытаний программного обеспечения «AlphaBI» версии 5.3.0 (далее – ПО), разработанного АО «БАРС Груп», и операционной системы специального назначения «Astra Linux Special Edition» РУСБ.10015-01 (очередное обновление 1.8) (далее – Astra Linux SE 1.8.0), разработанной ООО «РусБИТех-Астра».

2 Объект испытаний

2.1 Перечень компонентов, эксплуатировавшихся в ходе проведения данных испытаний, относящихся к ПО, представлен в Таблице 1.

Таблица 1 – Перечень компонентов, относящихся к ПО

Описание	Наименование	MD5	Источник
Файл архива, содержащий основные модули ПО	Bars.Alpha.Web-linux-x86_64-5.3.0.24324.zip	886f6f48b94eb3eb401f057c8472d90f	Сторона разработчика ПО
Файл архива, содержащий дополнительные модули ПО	Bars.Alpha.Mondrian-3.14.0.15_13.189.war	d73623e5e5dd378cd66c23329573c3aa	Сторона разработчика ПО
Файл архива, содержащий набор инструкций к инициализации ПО	Bars.Alpha.AdminGuide-5.3.0.24324.zip	a858327e2f30703297b59ae1b420b8d4	Сторона разработчика ПО

3 Ход испытаний

3.1 В ходе проведения настоящих испытаний были выполнены проверки корректности функционирования ПО в среде: Astra Linux SE 1.8.0, в объеме, указанном в Приложении 1.

3.2 Перечень используемых репозиторий приведен в Приложении 2.

3.3 Неофициальные репозитории ПО для указанных сред не эксплуатировались.

3.4 По информации от разработчика ПО не поддерживает работу с активным режимом ЗПС. Испытания проводились при отключенном режиме ЗПС.



3.5 Проверка корректности функционирования ПО в условиях ненулевого уровня конфиденциальности механизма мандатного разграничения доступа (далее – МРД) указанных сред не проводилась по причине отсутствия поддержки ПО соответствующей функциональности ОС. Информация об отсутствии упомянутой поддержки была заявлена стороной разработчика ПО.

4 Результаты испытаний

4.1 ПО корректно функционирует в среде: Astra Linux SE 1.8.0 .

5 Вывод

5.1 ПО и операционная система специального назначения «Astra Linux Special Edition» РУСБ.10015-01 (очередное обновление 1.8) совместимы, принимая во внимание информацию, содержащуюся в разделах 3, 4 и Приложении 2.

6 Состав рабочей группы и подписи сторон

6.1 Данный протокол составлен участниками рабочей группы:

Проканюк Д. С. – начальник сектора отдела технологической совместимости департамента развития технологического сотрудничества ДВиС ООО «РусБИТех-Астра»;

Лукашенко И. А. – инженер отдела технологической совместимости департамента развития технологического сотрудничества ДВиС ООО «РусБИТех-Астра».



Перечень проверок совместимости ПО и Astra Linux SE 1.8.0

№ п/п	Наименование проверки	Результат проверки ПО и Astra Linux SE 1.8.0
		с ядром ОС
		6.1.50-1-generic
1.	Установка ПО	Успешно
2.	Запуск, остановка выполнения ПО	Успешно
3.	Эксплуатация минимальной базовой функциональности ПО	Успешно
4.	Функционирование ПО в условиях низкого уровня целостности механизма МКЦ ОС	Успешно
5.	Функционирование ПО в условиях ненулевого уровня конфиденциальности механизма МРД ОС	Не проводилась
6.	Отсутствие нарушений требований подраздела 17.3 «Руководство по КСЗ Ч. 1»	Успешно
7.	Соответствие объектов ФС ОС дистрибутиву ОС при эксплуатации ПО	Успешно
8.	Удаление ПО	Успешно
9.	Функционирование ПО в условиях включённого механизма ЗПС ОС	Не проводилась
10.	Отсутствие нарушений требований подраздела 17.2 «Руководство по КСЗ Ч. 1»	Успешно



Инструкция по установке и удалению ПО в средах: Astra Linux SE 1.8.0

1 Используемые репозитории:

в Astra Linux SE 1.8.0:

- deb https://dl.astralinux.ru/astra/frozen/1.8_x86-64/1.8.0/repository-main/ 1.8_x86-64 main contrib non-free non-free-firmware
- deb https://dl.astralinux.ru/astra/frozen/1.8_x86-64/1.8.0/repository-devel/ 1.8_x86-64 main contrib non-free non-free-firmware
- deb https://dl.astralinux.ru/astra/frozen/1.8_x86-64/1.8.0/repository-extended/ 1.8_x86-64 main contrib non-free non-free-firmware
- deb [trusted=yes] <http://ftp.de.debian.org/debian> bullseye main

2 Установка ПО:

2.1 выполнить системные команды, действия:

```
sudo apt install ntp postgresql-15 openjdk-17-jre tomcat9 unzip zip nfs-kernel-server  
nginx haproxy nfs-common libgssapi-krb5-2 libicu72 libssl3 zlib1g libgdiplus mc ssl-  
cert
```

#Для установки etcd добавить в sources.list:

```
deb [trusted=yes] http://ftp.de.debian.org/debian bullseye main
```

```
sudo apt install etcd
```

#заменить директивы pool на блок директив (Здесь необходимо корректно указать данные вашего ntp):

```
sed -i '/pool\ 0/c server 0.ru.pool.ntp.org' /etc/ntp.conf
```

```
sed -i '/pool\ 1/c server 1.ru.pool.ntp.org' /etc/ntp.conf
```

```
sed -i '/pool\ 2/c server 2.ru.pool.ntp.org' /etc/ntp.conf
```

```
sed -i '/pool\ 3/c server 3.ru.pool.ntp.org' /etc/ntp.conf
```

#Рестарт ntp:

```
sudo systemctl restart ntp
```

```
sudo sed -e 's/#max_prepared_transactions.*/max_prepared_transactions = 100/' -
```



```
i /etc/postgresql/15/main/postgresql.conf
sudo sed -e "s/#listen_addresses.*/listen_addresses = '*'/" -i
/etc/postgresql/15/main/postgresql.conf
```

```
#В файле /var/lib/pgsql/11/data/pg_hba.conf
#Метод аутентификации поменять на "trust" в необходимых строках и произвести
ребут postgresql
sudo systemctl restart postgresql
```

```
#Конфигурирование postgresql
#Поменять пароль пользователя postgres
sudo passwd postgres
#В текущем примере из отдельной сессии пользователя postgres:
psql
postgres=# create user AlphaBl password '123' createdb;
postgres=# create database AlphaBase owner Alpha;
```

```
#Конфигурирование Tomcat
```

```
echo 'JAVA_OPTS="-Djava.awt.headless=true
-Djava.security.egd=file:/dev/./urandom -Xms512M -Xmx1024M -server -XX:
+UseZGC"' > /etc/default/tomcat9
#Вместо -Xmx1024M следует подобрать значение, соответствующее размеру
доступной памяти сервера.
```

```
systemctl restart tomcat9
systemctl status tomcat9
```

```
#Развертывание Mondrian
```

```
#Отредактировать файл WEB-INF/web.xml архива Bars.Alpha.Mondrian-  
{mondrian_version}.war через mc
#Изменить значение параметра DataSourceConfig:
```



```
{ <init-param>
<param-name>DataSourcesConfig</param-name>
<param-value>http://${BalancerHost}:${HaproxyPort}/mondrian/datasources</
param-value>
</init-param> }
```

#Изменить значение параметра AlphaApiKey (прим. данный параметр не влияет ни на что, может быть вписано любое значение как числовое так и символьное):

```
{<init-param>
<param-name>AlphaApiKey</param-name>
<param-value>${MondrianApiKey}</param-value>
</init-param>}
```

#Отредактировать файл WEB-INF/classes/mondrian.properties
mondrian.util.memoryMonitor.percentage.threshold=99

Переместить файл /tmp/Bars.Alpha.Mondrian-\${mondrian_version}.war в каталог приложений Tomcat, изменив его название на mondrian.war

```
sudo -s
systemctl restart tomcat9
systemctl status tomcat9
```

```
#Сервер файлового хранилища
#Конфигурирование
mkdir /var/lib/alphabi_file_storage
chown nobody:nogroup /var/lib/alphabi_file_storage/
echo "/var/lib/alphabi_file_storage
*(rw,async,all_squash,anonuid=65534,anongid=65534,no_subtree_check)" >>
/etc/exports
exportfs -ra
```

```
systemctl restart nfs-kernel-server
```



#Сборка и запуск кластера Etcd

#Отредактировать файл /etc/default/etcd , указав следующие значения:

```
ETCD_NAME="etcd_alphabi_1"
ETCD_LISTEN_CLIENT_URLS="http://${EtcdHost_1}:${EtcdPort},http://localhost:${EtcdPort}"
ETCD_ADVERTISE_CLIENT_URLS="http://${EtcdHost_1}:${EtcdPort}"
ETCD_LISTEN_PEER_URLS="http://${EtcdHost_1}:${EtcdPeerPort}"
ETCD_INITIAL_ADVERTISE_PEER_URLS="http://${EtcdHost_1}:${EtcdPeerPort}"
ETCD_INITIAL_CLUSTER_TOKEN="alphabi"
ETCD_INITIAL_CLUSTER="etcd_alphabi_1=http://${EtcdHost_1}:${EtcdPeerPort}"
ETCD_INITIAL_CLUSTER_STATE="new"
ETCD_AUTO_COMPACTION_RETENTION="24"
```

systemctl start etcd

#Конфигурирование последующих узла кластера Etcd (прим. использовался только один узел кластера в данном примере)

```
ETCDCTL_API=3 etcdctl --endpoints ${EtcdHost_1}:${EtcdPort} member add
etcd_alphabi_${Current} --peer-urls http://${EtcdHost_Current}:${EtcdPeerPort}
#Отредактировать файл /etc/default/etcd , указав следующие значения:
```

```
ETCD_NAME="etcd_alphabi_${Current}"
ETCD_LISTEN_CLIENT_URLS="http://${EtcdHost_Current}:${EtcdPort},http://localhost:${EtcdPort}"
ETCD_ADVERTISE_CLIENT_URLS="http://${EtcdHost_Current}:${EtcdPort}"
ETCD_LISTEN_PEER_URLS="http://${EtcdHost_Current}:${EtcdPeerPort}"
ETCD_INITIAL_ADVERTISE_PEER_URLS="http://${EtcdHost_Current}:${EtcdPeerPort}"
ETCD_INITIAL_CLUSTER_TOKEN="alphabi"
ETCD_INITIAL_CLUSTER="${ETCD_INITIAL_CLUSTER}"
ETCD_INITIAL_CLUSTER_STATE="existing"
```



```
ETCD_AUTO_COMPACTION_RETENTION="24"
```

#Где `{ETCD_INITIAL_CLUSTER}` - это значение `ETCD_INITIAL_CLUSTER`, выведенное командой `etcdctl member add` с прошлого шага;

`ETCD_AUTO_COMPACTION_RETENTION` - параметр, отвечающий за автоматическое сжатие пространства ключей

```
systemctl start etcd
```

```
ETCDCTL_API=3 etcdctl user add root:123
```

#Вместо '123' следует подставить новый пароль

#Создать пользователя в etcd и назначить этому пользователю права на диапазон ключей, выделенный для Alpha BI

```
ETCDCTL_API=3 etcdctl role add {EtcUser}
```

```
ETCDCTL_API=3 etcdctl role grant-permission {EtcUser} --prefix=true readwrite $
{EtcPrefix}
```

```
ETCDCTL_API=3 etcdctl user add {EtcUser}:{EtcPassword}
```

```
ETCDCTL_API=3 etcdctl user grant-role {EtcUser} {EtcUser}
```

```
ETCDCTL_API=3 etcdctl auth enable
```

#Установка

```
rm /etc/nginx/sites-available/default
```

Создать файл `/etc/nginx/sites-available/alphabi.conf` со следующим содержимым(прим. параметры `{PathToCertFile}` и `{PathToCertKeyFile}` - пути до файлов сертификата. В примере использовался самописный сертификат через `ssl-cert`):

```
server {
    listen 80 default_server;
```




```
location / {
    return 301 https://$host$request_uri;
}
}

server {
    listen 443 ssl http2;

    ssl_certificate ${PathToCertFile};
    ssl_certificate_key ${PathToCertKeyFile};
    ssl_session_timeout 1d;
    ssl_session_cache shared:MozSSL:10m;
    ssl_session_tickets off;

    # modern configuration
    ssl_protocols TLSv1.2 TLSv1.3;
    ssl_prefer_server_ciphers off;

    add_header Strict-Transport-Security "max-age=63072000" always;

    # OCSP stapling
    ssl_stapling on;
    ssl_stapling_verify on;

    # replace with the IP address of your resolver
    resolver 127.0.0.1;

    location / {
        proxy_pass http://127.0.0.1:${HaproxyPort};
        include proxy_params;
    }
}
```



#Создать или отредактировать файл /etc/nginx/proxy_params, приведя его содержимое к следующему виду

```
proxy_redirect off;
proxy_set_header Host $http_host;
proxy_set_header X-Real-IP $remote_addr;
proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
proxy_set_header Connection "";
proxy_http_version 1.1;
proxy_hide_header x-aspnet-version;
proxy_hide_header x-powered-by;
proxy_hide_header x-aspnetmvc-version;
client_max_body_size 512m;
client_body_buffer_size 8m;
proxy_connect_timeout 120;
proxy_send_timeout 1800;
proxy_read_timeout 1800;
proxy_buffer_size 1024k;
proxy_buffers 32 1024k;
proxy_temp_file_write_size 1024k;
proxy_intercept_errors on;
```

#Создать символическую ссылку:

```
sudo ln -s /etc/nginx/sites-available/alphabi.conf
/etc/nginx/sites-enabled/alphabi.conf
```

```
systemctl restart nginx
```

#Конфигурационный файл /etc/haproxy/haproxy.cfg привести к следующему виду:

```
global
    log /dev/log local0
```



```
log /dev/log local1 notice
chroot /var/lib/haproxy
stats socket /run/haproxy/admin.sock mode 660 level admin
stats timeout 30s
user haproxy
group haproxy
maxconn 20000
daemon

#Default SSL material locations
ca-base /etc/ssl/certs
crt-base /etc/ssl/private

#Default ciphers to use on SSL-enabled listening sockets.
#For more information, see ciphers(1SSL). This list is from:
# See: https://ssl-config.mozilla.org/#server=haproxy&server-version=2.0.3&config=intermediate
ssl-default-bind-ciphers ECDHE-ECDSA-AES128-GCM-SHA256:ECDHE-RSA-AES128-GCM-SHA256:ECDHE-ECDSA-AES256-GCM-SHA384:ECDHE-RSA-AES256-GCM-SHA384:ECDHE-ECDSA-CHACHA20-POLY1305:ECDHE-RSA-CHACHA20-POLY1305:DHE-RSA-AES128-GCM-SHA256:DHE-RSA-AES256-GCM-SHA384
ssl-default-bind-ciphersuites
TLS_AES_128_GCM_SHA256:TLS_AES_256_GCM_SHA384:TLS_CHACHA20_POLY1305_SHA256
ssl-default-bind-options ssl-min-ver TLSv1.2 no-tls-tickets

defaults
log global
mode http
timeout connect 60s
timeout client 30m
timeout server 30m
timeout http-request 5m
```



```
timeout queue 5m
timeout http-keep-alive 10s
timeout check 30s
errorfile 400 /etc/haproxy/errors/400.http
errorfile 403 /etc/haproxy/errors/403.http
errorfile 408 /etc/haproxy/errors/408.http
errorfile 500 /etc/haproxy/errors/500.http
errorfile 502 /etc/haproxy/errors/502.http
errorfile 503 /etc/haproxy/errors/503.http
errorfile 504 /etc/haproxy/errors/504.http
compression algo gzip
compression type text/html text/plain text/xml text/css application/javascript
application/json
option forwardfor
option http-pretend-keepalive

frontend FE-AlphaBI-http
bind 127.0.0.1:${HaproxyPort}
mode http
option httplog
option dontlognull
option log-separate-errors
option forwardfor header X-Real-IP
backlog 4096
maxconn 20000

default_backend BE-BI-ALPHA

frontend FE-Mondrian-http
bind ${BalancerHost}:${MondrianPort}
mode http
option httplog
option dontlognull
option log-separate-errors
```



```
option forwardfor header X-Real-IP
backlog 4096
maxconn 20000
```

```
default_backend BE-MONDRIAN
```

```
backend BE-BI-ALPHA
```

```
mode http
balance leastconn
cookie NodeID insert indirect nocache
log global
option log-health-checks
fullconn 20000
http-check disable-on-404
option httpchk HEAD /cc/
server alpha-bi-01 ${WebAppHost_1}:${WebAppPort} check port $
{WebAppPort} inter 5s weight 32 maxconn 1000 cookie alphabi-01
server alpha-bi-N ${WebAppHost_N}:${WebAppPort} check port $
{WebAppPort} inter 5s weight 32 maxconn 1000 cookie alphabi-N
```

```
backend BE-MONDRIAN
```

```
mode http
balance leastconn
cookie NodeID insert indirect nocache
log global
option log-health-checks
fullconn 20000
http-check disable-on-404
option httpchk HEAD /${MondrianAppName}
server olap-01 ${MondrianHost}:${MondrianPort} check port $
{MondrianPort} inter 5s weight 32 maxconn 1000 cookie olap-01
```

#{WebAppPort} - порт, на котором работает AlphaBI \${WebAppHost_1} - адрес



сервера, на котором развернута AlphaBI. Вместо строки `server alpha-bi-N $`
`{WebAppHost_N}:{WebAppPort} check port ${WebAppPort} inter 5s weight 32`
`maxconn 1000 cookie alphabi-N` добавить нужное количество строки по
количеству серверов с развернутым приложением AlphaBI, вместо N указать
порядковый номер.

```
sudo haproxy -f /etc/haproxy/haproxy.cfg -c  
sudo systemctl restart haproxy
```

Скопировать дистрибутив AlphaBI на сервер. `Bars.Alpha.Web-linux-x86_64-${`
`{version}.zip`.

```
sudo useradd -m --system alphabi
```

```
sudo -s
```

```
mkdir -p /opt/alphabi /var/log/alphabi  
unzip /tmp/Bars.Alpha.Web-linux-x86_64-${version}.zip -d /opt/alphabi  
chown -R alphabi /opt/alphabi /var/log/alphabi  
mkdir -p /var/tmp/alphabi  
chown -R alphabi /var/tmp/alphabi  
mkdir -p /mnt/nfs/alphabi_file_storage  
echo '${FsHost}:/var/lib/alphabi_file_storage /mnt/nfs/alphabi_file_storage/ nfs  
rw,sync,intr 0 0' >> /etc/fstab  
mount -a
```

#Заменить `${FsHost}` на адрес NFS-сервера

```
sudo -s  
mkdir -p /var/lib/alphabi_file_storage/  
chown alphabi:alphabi /var/lib/alphabi_file_storage/
```



#Создать файл /etc/systemd/system/alphabi.service следующего содержания:

[Unit]

Description=AlphaBI: Web application

After=syslog.target network.target

[Service]

User=alphabi

Environment=ASPNETCORE_ENVIRONMENT=Production

Environment=ASPNETCORE_URLS=http://*:\${WebAppPort}/

Environment=ASPNETCORE_URL_BASE=\${ExternalURLPathPrefix}

Environment=ASPNETCORE_USE_XFORWARDEDFOR=true

WorkingDirectory=/opt/alphabi

ExecStart=/opt/alphabi/bin/Bars.Alpha.Web

RestartSec=10

Restart=always

[Install]

WantedBy=multi-user.target

#Создать /opt/alphabi/alpha.config следующего содержания: (alpha.config строится на основе alpha.release.config; alpha.release.config содержит комментарии):

```
{
  "ControlCenterConfig": {
    "Password": "${AlphaControlCenterPassword}"
  },
  "DatabaseConfig": {
    "ServerAddress": "${PostgresHost}",
    "ServerPort": ${PostgresPort},
    "User": "${PostgresUser}",
    "Password": "${PostgresPassword}",
    "AdditionalParams": "",
    "DatabaseName": "${PostgresDatabase}",
    "UserCommandTimeoutSeconds": 30
  },
  "Cluster": {
```



```

    "IsClustered": true,
    "Etcd": {
      "Host": "${EtcdHost_1}",
      "Port": ${EtcdPort},
      "UserName": "${EtcdUser}",
      "Password": "${EtcdPassword}",
      "KeyPrefix": "${EtcdPrefix}"
    }
  },
  "OlapConfig": {
    "XmlaEndpoint": "http://${BalancerHost}:${MondrianPort}/${MondrianAppName}/xmla",
    "ApplicationBaseUrl": "http://${BalancerHost}:${WebAppPort}${ExternalURLPathPrefix}",
    "ApiKey": "${MondrianApiKey}"
  },
  "FileStorageConfig": {
    "RootPath": "/mnt/nfs/alphabi_file_storage"
  },
  "LogConfig": {
    "FileLogging": "info",
    "DbLogging": "info",
    // Путь до директории с логами. По умолчанию директория .logs в рабочей
    // директории приложения
    "LogDirectory": "/var/log/alphabi"
  },
  "TempDir": {
    "Path": "/var/tmp/alphabi",
    "Clear": true
  },
  "ApplicationInfo": {
    "PublicUrl": "${ExternalURL}"
  },
  "ReportGenerator": {

```




```
"JavaExecutable": "${JavaExecutable}"  
},  
}
```

```
chown alhab1 /opt/alhab1/alpha.config  
su - alhab1 -s /bin/bash  
cd /opt/alhab1  
./bin/Bars.Alpha.Web install ${Password}
```

```
systemctl start alhab1  
systemctl enable alhab1
```

3 Удаление ПО:

3.1 выполнить системные команды, действия:

```
sudo apt autoremove --purge ntp postgresql-15 openjdk-17-jre tomcat9 unzip zip  
nfs-kernel-server nginx haproxy nfs-common libgssapi-krb5-2 libicu72 libssl3 zlib1g  
libgdiplus mc ssl-cert
```

```
sudo rm -rf /opt/'файлы ПО'
```



Перечень используемых сокращений

«Руководство по КСЗ Ч. 1» – документ «Операционная система специального назначения «Astra Linux Special Edition». Руководство по КСЗ. Часть 1» РУСБ.10015-01 97 01-1;

Astra Linux SE 1.8.0 – операционная система специального назначения «Astra Linux Special Edition» РУСБ.10015-01 (очередное обновление 1.8);

ДВиС – дирекция внедрения и сопровождения;

ЗПС – замкнутая программная среда;

КСЗ – комплекс средств защиты;

МКЦ – мандатный контроль целостности;



МРД – мандатное управление доступом;

ОС – операционная система;

ПО – программное обеспечение «AlphaBI» версии 5.3.0.

Идентификатор документа 96a4620f-aa2b-4836-b374-a720a0603f73

Документ подписан и передан через оператора ЭДО АО «ПФ «СКБ Контур»

Подписи отправителя:	 ООО "РУСБИТЕХ-АСТРА" Проканюк Дмитрий Сергеевич	 Не приложена при подписании	Доверенность: рег. номер, период действия и статус Сертификат: серийный номер, период действия 043C5A7100B6B007A24D9A5E4F 91BFE299 с 10.11.2023 09:42 по 10.11.2024 09:42 GMT+03:00	Дата и время подписания 27.05.2024 16:38 GMT+03:00 Подпись соответствует файлу документа
-------------------------	--	---	--	---