Forcepoint Behavioral Analytics

3.4.2 LIMITED AVAILABILITY INSTALLATION GUIDE

Publish Date: March 28, 2022

F23-09-02-03282022

Legal Notice

Attributions

All terms mentioned in this document that are known to be trademarks or service marks have been appropriately capitalized. The company cannot attest to the accuracy of this information. Use of a term in this document should not be regarded as affecting the validity of any trademark or service mark.

Linux® is the registered trademark of Linus Torvalds in the U.S. and other countries.

Document Conventions

The following typographic conventions are used in this guide:

Typography

Format	Description
Bold font	Used to identify Graphical User Interface (GUI) elements, buttons, fields, and list labels. Example: Type your IP address in the **ip address** field and click **OK**.
Italic font	Used to identify book titles or words that require emphasis. Example: Read the *User's Guide*.
Monospaced font	Used to identify names of commands, files, and directories. Example: Use the `ls -a` command to list all files.
Monospaced bold font	When inline, this is used to identify text that users need to type. Example: Type **`SYSTEMHIGH`** in the **Network** field.
Shaded monospaced font	Used to identify screen output. Example: A network device must exist; otherwise, the following warning message displays `Warning: device [DEVICE] is not a valid network device`
Shaded monospaced bold font	Used to identify text that users need to type. Example: Specify your network configuration. Type: **`$ sudo ip addr show`**

This guide makes use of the following elements:

Note
Contains important information, suggestions or references to material covered elsewhere in the guide.

Tip
Provides helpful suggestions or alternative methods to perform a task.

Warning
Alerts you to an activity that may cause permanent loss of data or product functionality. Failure to heed a warning could result in permanent consequences to your data or system.

Caution
Alerts you to anything that could result in a security breach or temporary loss of data or product functionality. You may also see a caution when a particular action may have an adverse impact that is not readily apparent.

Important
Highlights critical tasks, information or actions that may be damaging to your system or security.

CONTENTS

List of Figures

List of Tables

Install Guide for Forcepoint Behavioral Analytics 3.4.2 Limited Availability

OVERVIEW

This Forcepoint Behavioral Analytics (FBA) Installation manual guides technical FBA users through a complete installation of an FBA deployment. This guide includes step-by-step instructions for installing FBA via Ansible® and Jenkins®. This document covers system architecture, required software installation tools, and finally a step-by-step guide for a complete install.

The System Architecture section shows how data moves throughout software components, as well as how third party software is used for key front and back-end functionalities.

The Installation Components section elaborates on important pre-installation topics. In preparation for the initial installation setup, high-level topics regarding Jenkins and Ansible, and the tools FBA utilizes to facilitate installation commands are described. Additionally, it is strongly recommend that the *Forcepoint Behavioral Analytics Hardening Guide* is followed (available through Professional Services) to ensure the system is set up with security best practices.

Additionally, step-by-step instructions for using Ansible to initialize the Jenkins CI/CD server to install each required software component is included.

An addendum is included for additional components that can optionally be installed.

Go to the Downloads page and navigate to Forcepoint Behavioral Analytics to find the downloads for Forcepoint Behavioral Analytics.

Important
For those installs that are NOT to be used for Dynamic Data Protection (DDP) setups and for those environments that require running in a "root-less" mode, please read the Appendix before proceeding with the install.

PLATFORM OVERVIEW

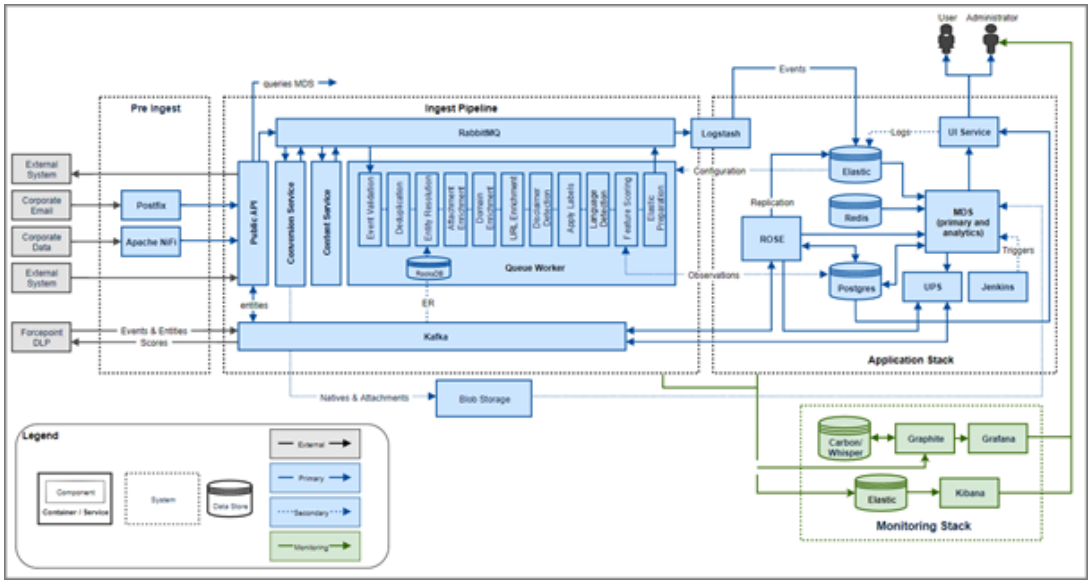

Figure 1. Component Architecture

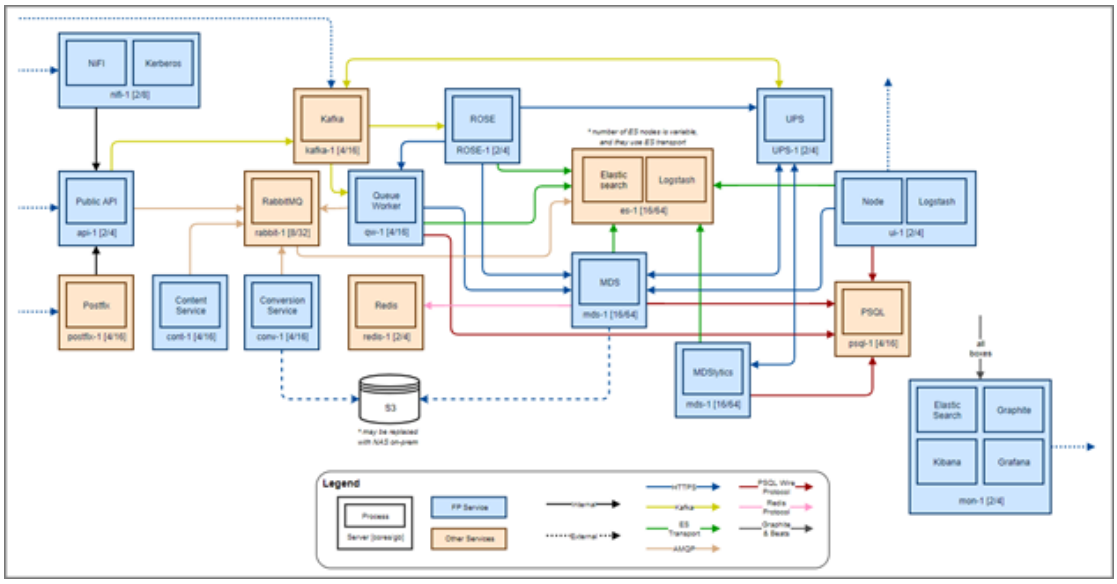

Figure 2. Physical Architecture

INSTALLATION COMPONENTS OVERVIEW

- Host OS
  - Forcepoint requires a Red Hat® 7.9 host-based Operating System for the FBA platform to be installed. **A minimum kernel version of 3.10.0, as well as CentOS 7.9 or RHEL 7.9, are required.** The minimal versions of either CentOS or RHEL should be used. Please note, other heavier installs should not be used as there might be Red Hat Package Managers (RPM) that cause conflicts with required RPMs and versions of those RPMs. CentOS and RHEL versions 7.8 and below are no longer supported and are blocked through the install process.
- Docker Containers
  - Continued in FBA 3.4.2 Rootless is the use of Docker version 20.10.6 to house the services and applications. Docker is included in the installation media and installed through the provided install scripts. All ports mentioned below are being mapped from the host OS to the container. Each host OS contains a single container running a minimal CentOS 7.9 container image.
- Security
  - Please refer to the *Hardening Guide* for best security practices. Forcepoint recommends using commonly accepted network security practices to restrict access to the FBA infrastructure. For instance, creating rules in IPTables, or implementing a network firewall that only allows the access defined in the ports list below.

- Port Map

Table 1. Port Map

Service Name	Host	Port	Consumers
SSH	All	2222	All containers use 2222 for SSH
Redis	`redis`	6379	UI
Graphite	`mon`	2003	All
Grafana	`mon`	443	Administrator Workstation
Jenkins	`jenkins`	8080 8443 80	All, Administrator Workstation
Vault	`jenkins`	8200 8201 8300 8301	All, Administrator Workstation
Kafka	`kafka`	9092-9095	API, Rose
Kafka Manager	`kafka`	9000	Administrator Workstation
Postgres	`postgres`	5432	Conversion, Rose, Master Data Service, Queue Worker, UI
NiFi	`nifi`	88/464/749	kerberos
NiFi	`nifi`	1521	Oracle Database Connection
NiFi	`nifi`	443/8443	Administrator Workstation
RO-API	`api`	9000	External Data Sources, RabbitMQ
RO-API	`api`	9001	Administrator Workstation
RO-Conv	`conv`	9080	RabbitMQ
RO-Conv	`conv`	9081	Administrator Workstation
RO-Cont	`cont`	9700	RabbitMQ, ES
RabbitMQ	`rabbit`	4369	Administrator Workstation
RabbitMQ	`rabbit`	15672	Administrator Workstation
ro-qw	`qw`	9090	RabbitMQ

Service Name	Host	Port	Consumers
ro-qw	`qw`	9091	Administrator Workstation
UI	`ui`	80 443	Users
Elasticsearch	`es`	9200	UI, Jenkins, ES, MDS, API, Conv, QW
Elasticsearch	`es`	9201	Administrator Workstation
Elasticsearch	`es`	9300-9400	Elasticsearch
ro-mds	`mds` `mdslytics`	8080	UI, Jenkins, MDS
ro-mds	`mds` `mdslytics`	8081	Adminitrator Workstation
ro-rose	`rose`	9500	API, Postgres, Nifi, QW, UPS
ro-rose	`rose`	9501	Administrator Workstation
ro-ups	`ups`	9600	MDS
ro-ups	`ups`	9601	Administrator Workstation
OpenVPN	`vpn`	1194	External Clients

INSTALLATION REQUIREMENTS

A minimum kernel version of 3.10.0, as well as CentOS 7.9 or RHEL 7.9, are required. This is due to the packages needed in order to correctly run the docker service on the host (specifically, the `shadow-utils` package for the ability to run `newuidmap` and `newgidmap`).

Warning

While it is possible to manually install the required packages and use and lower version of CentOS or RHEL, it is not supported nor is it recommended and can lead to installation and runtime errors.

External access to ports 8080 and 8443 on the Jenkins host is critical during installation as great care has been given to automate the installation process through Jenkins jobs. While it is technically feasible to bypass the Jenkins jobs and run the install solely via Ansible playbooks, this is not recommended as permissions and least privilege are put at risk. Please contact Forcepoint Support for further information.

Installation Facilitators

Jenkins

Jenkins is an open-source automation server that helps to automate the non-human part of continuous delivery. This is the primary way in which Forcepoint installs the FBA software.

Makeself

The FBA installation media is packaged using self-extractable archives through an open-source solution named makeself (`makeself.io`). This allows for the FBA installation bundle to run and be pre-installed on a primary host (Jenkins) and extract the required packages into the proper directories under the provisioned user with limited privileges. During this phase of the install process, the required Docker binaries and required scripts are installed and ran.

The steps completed by makeself are as follows:

- untar files
- run `pre-install.sh`
  - Install required host-level RPMs.
  - Verify that the required directories are created.
  - Verify that docker can be installed on each host.
  - Install docker on each host.
  - Run docker on each host.
  - Setup the yum repo and RPMs on mounted volume inside the container.
  - Install RPMs needed to run the install inside the Jenkins container.

Ansible

Ansible is an IT automation tool that can configure systems, deploy software, and orchestrate more advanced IT tasks such as continuous deployments or zero downtime rolling updates. Ansible playbooks are used to incrementally install the separate components of an FBA instance.

File Format: YAML

Ansible uses YAML because it is easier for humans to read and write than other common data formats, like XML or JSON. Further, there are libraries available in most programming languages for working with YAML.

Playbooks

- Playbooks are the basis for simple configuration management and a multi-machine deployment system that is well suited to deploy complex applications.
- Playbooks can declare configurations, and they can also orchestrate steps of any manual ordered process, even as different steps must bounce back and forth between sets of machines in particular orders. They can launch tasks synchronously or asynchronously.
- Individual “Tasks” Make Up a role or playbook. A “Playbook” is comprised of tasks and roles (Figure 3).

```
- hosts: webservers
 remote_user: root

 tasks:
 - name: ensure apache is at the latest version
 yum: name=httpd state=latest
 - name: write the apache config file
 template: src=/srv/httpd.j2 dest=/etc/httpd.conf

- hosts: databases
 remote_user: root

 tasks:
 - name: ensure postgresql is at the latest version
 yum: name=postgresql state=latest
 - name: ensure that postgresql is started
 service: name=postgresql state=started
```

Figure 3. Example Ansible Playbook

INSTALLATION PROCEDURES

Things to consider before beginning the installation of FBA.

- All infrastructure must be provisioned beforehand.
  - All hosts as needed for the size of the deployment.

    Note

    Every major component in the FBA technical stack runs on its own host.

  - Appropriate networking considerations must be made.
  - Appropriate amount of local disk storage.
  - Network File Storage (NFS) shared storage or S3 (this will depend on the location of the install: on-prem or an Amazon Web Services (AWS)).
- The minimum kernel version must be 3.10.0.

  Note

  Disabling swap on all hosts is highly recommended, and at a minimum, this must be done on the Elasticsearch hosts. This can be done by running the following command on all nodes and then removing any mount points for swap in `/etc/fstab`.

  ```
  swapoff -a
  ```

  Tip

  If installing under VMware ™ the package open-vm-tools should be used for better VM support.

- Python® version 2.7 must be used within the docker containers. Version 2.7.5 is included in the latest version of the FBA installer at the time of publication.
- All hosts must have SSH enabled and reachable from the provisioning Ansible host (Jenkins) via `/etc/hosts` or `DNS`.
- `$FBA_USER` is used below in place of the provisioned user created for installation and runtime of the FBA software.

  Note

  This user must have full passwordless sudo permissions (passwordless SSH will be set up as part of the installation process).

- `$FBA_DIR` is the directory that the installer will be run from and must be owned by the `$FBA_USER`.
  - Required file paths are as follows:
    - `$FBA_DIR`
      - Base install directory.
      - Best practice is to have this mounted on a separate partition from the root OS.

    - `$FBA_DIR/data`
        - Data volumes will be mounted here.
        - Best practice is to have this mounted on a separate partition from the root OS.
- Optional file paths:
    - `$FBA_DIR/data/nfs`
        - For use when not using S3 for Large Object (LOB) storage.
        - This is required to be mounted to the host under this directory for use within the container.
        - The NFS server must be configured with exports set to include `all_squash` and the `anonuid` and `anongid` set to the `$FBA_USER`'s `uid` and `gid`.
    - `$FBA_DIR/var/log`
        - Best practice is to have this mounted on a separate partition from the root OS and the `$FBA_DIR`.
- The FBA installation and configuration is Ansible based. The following requirements must be met:
    - The `hosts` file must be accurate. A template of the `hosts` file is included.
        - This will replace the file `/etc/hosts`.
    - The `ansible-all` file must be accurate and tailored to any site-specific overrides if necessary. A template of the `ansible-all` file is included.
        - The `ansible-all` file must be copied to the `/etc/ansible/group_vars/all` inside the container.
    - The `ansible-hosts` file must be accurate and include all hosts in their appropriate groups. A template of the `ansible-hosts` file is included.
        - The ansible-hosts file must be copied to the `/etc/ansible/hosts` file inside the container.
    - Host machine running ansible playbooks (Jenkins server) must have ssh access to all hosts in the `/etc/ansible/hosts` inventory file.
- All commands are assumed to be run on a fully updated CentOS 7.9 or RHEL 7.9 host.
- Escalated privileges are required for the installation.
    - Installation must be done using the `sudo` user and not the `root` user.
    - Depending on security policies, for ease, the sudoers file should be updated to allow for passwordless sudo usage.
    - For the full allow list, refer to the Installation Appendix in this document.
- All remaining actions will be performed on the Jenkins host.
- Once the Jenkins server is initialized, the FBA platform is deployed using the Continuous Delivery server.

GETTING STARTED

Download the Forcepoint Behavioral Analytics installer media

1. Obtain the FBA installer from the Forcepoint Support Site:

   `https://support.forcepoint.com`

2. Untar the installer tarball under the `$FBA_DIR`

   ```
   tar xf FBA-342.tar.gz
   ```

Create and Configure the Hosts and Ansible Files

To accommodate custom directory installations, the source custom directory in the following steps will be referred to as the `$FBA_DIR`. This directory is the basis for all directories volume mounted within the container. The ideal location for this directory is the home directory of the provisioned user. For example `/home/$FBA_USER = $FBA_DIR`. The `$FBA_DIR` must be owned by the $FBA_USER. The $FBA_DIR is mapped to `/` when mounted inside the containers.

There are three system config files that must be created with care on the Jenkins host in order for the install and runtime processes to work successfully. Templates for these files are included in the installer tarball.

- `$FBA_DIR/hosts`
  - Use: Operating system file that translates hostnames or domain names to IP addresses.
  - Template name: `hosts`
- `$FBA_DIR/ansible-hosts`
  - Use: Config file used by Ansible for a list of hosts and groupings of hosts being managed.
  - Template name: `ansible-hosts`
- `$FBA_DIR/ansible-all`
  - Use: The top-level setting of variables used in the Ansible playbooks.
  - Template name: `ansible-all`

Using the provided template files as a starting point for these files is strongly recommended. To use the provided template files, complete the following steps:

1. Configure `/$FBA_DIR/hosts`

   Using the command in the example below will update the example file with the updated hostnames. The IP addresses will need to be updated to reflect the deployment. The example file is based on a minimal deployment and will need to be adjusted for the actual hosts in the deployment. i.e. additional ES nodes must be added manually.

   ```
   sed -i 's/xxxxx/change_me/g'
   hosts Example exert: hosts
   ##########################################
   127.0.0.1 localhost localhost.localdomain localhost4 localhost4.localdomain4 ::1
   localhost localhost.localdomain localhost6 localhost6.localdomain6
   10.55.10.110 api-xxxxx
   10.55.10.106 conversion-xxxxx
   10.55.10.105 jenkins-xxxxx
   ```

```
10.55.10.120 kafka-xxxxx
10.55.10.122 es1-xxxxx
10.55.10.124 es2-xxxxx
10.55.10.136 es3-xxxxx
10.55.10.137 mds-xxxxx
10.55.10.138 mdslytics-xxxxx
##########################################
```

2. Create and configure `/$FBA_DIR/ansible-hosts`

   Below is a command to use the template `ansible-hosts` file.

   - Perform a search and replace command using `sed`.
   - Copy the updated file to the correct location.

   The find and replace command will change `xxxxx` to the text that is in the `change_me` field. Make sure to change the `change_me` in the example below before running the command. It will be visible to users, so an abbreviation of the customer's name or an intuitive substitute for the customer's name is recommended.

   Note
   The example file is based on a minimal deployment and will need to be adjusted for the actual hosts in the deployment. i.e additional ES nodes must be added manually.

```
sed -i 's/xxxxx/change_me/g' ansible-hosts
Example excerpt: ansible-hosts
##########################################
[api]
api-xxxxx
[ca]
ro-root-ca ansible_host=jenkins-xxxxx
[content]
cont-xxxxx
[conversion]
conv1-xxxxx
conv2-xxxxx
[curator]
curator-xxxxx ansible_host=jenkins-xxxxx
[es]
es1-xxxxx
es2-xxxxx
es3-xxxxx
##########################################
```

3. Create and configure `$FBA_DIR/ansible-all`

   There are two example files provided for the `ansible-all` file:

   - AWS install
   - On-Prem install

   Choose the version based on the install location.

   Tip
   This file is extremely important and many errors in the install process are commonly due to missing variables or typos in this file.

   Important

   All of the `xxxxx` fields in this file will need to be manually modified as they are specific to the environment being created.

```
# AWS Install Version
cp ansible-all.aws.example ansible-all

# On-Prem Install Version
cp ansible-all.on-prem.example ansible-all

Example exert: ansible-all
##########################################
##offline install

yum_repo_epel_enabled: "{{ epel_repo_enable }}"
yum_repo_sslverify: "0"
ueba_offline_install: true

##environment name (domain)
ro_env: xxxxx
domain: "{{ domain_name }}"
tld: internal
domain_name: "ro.{{ tld }}"
##########################################
...
##########################################
### Jenkins secret key info, which is passed in and used for the deploy jobs
# NOTE: by default, the pub/priv key method is uncommented
# vars:
# ueba_jk_id: - ID of secret in jenkins, needs to match below
# ueba_ansible - name of the secret
# user - user var in secret key, defaults to ro_user above
# private_key: - private key on jenkins machine
# public_key: - public key on jenkins machine
# password: - if using a password, though not recommended
###
```

```
ueba_ansible:
user: "{{ ro_user }}"
private_key: |
-----BEGIN OPENSSH PRIVATE KEY-----
xxxxx
xxxxx
xxxxx
xxxxx
xxxxx
-----END OPENSSH PRIVATE KEY-----
public_key: "ssh-ed25519 xxxxxx"
# password: "xxxxxx"
########################################
```

Generate and Push SSH Keys to all Hosts

1. Generate an SSH key pair.

   It is recommended to use passwordless ssh key authentication. To create the keys run the example below as the `$FBA_USER`:

   ```
   ssh-keygen -t ed25519
   ```

2. Copy the SSH public key to all hosts defined in the `hosts` file.

   A script has been provided under `scripts/SSH_key_copy.sh` to allow the key generated above to be copied to all hosts in the `hosts` file.

   Important

   The script assumes there is a common password used for all of the hosts.

   To run the script, perfom the following:

   ```
   #1 Ensure permissions are set so that the script is executable
   chmod +x SSH_key_copy.sh

   #2 Ensure sshpass is installed on the system which sshpass

   #2a If not installed
   sudo yum install sshpass

   #3 Run the script and enter the password when prompted
   bash SSH_key_copy.sh
   ```

   Tip

   If the password contains an exclamation point (!), the shell script will attempt to interpret the the code prematurely and error out. If this occurs, manually change the script to use single quotes and the string of the password instead.

NFS Setup (On-Prem Installations Only)

As the final installation step, please ensure that NFS is installed and running on all of the relevant hosts, refer to Table 1.2. With the docker architecture, it is assumed that all relevant NFS installation/configuration is done prior to the creation of the containers; the NFS volumes will then be mounted into the container and used appropriately.

The default location for NFS storage is: `/$FBA_DIR/data/nfs`

Table 1.2. Relevant FBA Hosts

Service Type	Default Name
Conversion Service	`conv1-{{hostname}}`
Conversion Service	`conv2-{{hostname}}`
NiFi	`nifi-{{hostname}}`
Elasticsearch	`es1-{{hostname}}`
Elasticsearch	`es2-{{hostname}}`
Elasticsearch	`es3-{{hostname}}`
Master Data Service	`mds-{{hostname}}`
Master Data Service	`mdslytics-{{hostname}}`
UI	`ui-{{hostname}}`

Note

Any services in the above list that have been scaled out horizontally (i.e. additional elasticsearch nodes) will also need NFS set up.

Run the Forcepoint Behavioral Analytics installer

Note

This script should be run as the `$FBA_USER` within the `$FBA_DIR`

1. Set the FBA installer to be executable.

```
chmod +x Forcepoint-UEBA-3.4.2-rootless.bin
```

2. Extract the FBA installer.

```
bash Forcepoint-UEBA-3.4.2-rootless.bin
```

3. The installer will first run the pre-install scripts which will prep the Jenkins host to run parallel and install Docker across the hosts.

Tip
Monitor the logs from the pre-install script for updates on status and to ensure there are no error

4. Once the pre-install completes successfully you will be prompted to run the following:

```
docker exec jenkins-{stack-name}-docker su - centos -c 'ansible-playbook
/usr/share/ro-ansible/jenkins-init.yml'
```

When this script completes, the Jenkins UI will be fully functional.

Deploy FBA from Jenkins

1. Navigate to the Jenkins web-based service in a browser.

Tip

The hostname can be reached by hostname, FQDN, or IP. For example:

`http://jenkins-customer.domain.com:8080`

`http://jenkins-customer:8080`

`http://10.0.0.100:8080`

2. Login to Forcepoint Continuous Delivery Server in Jenkins (Figure 4).
   a. Default credentials:

   Username: forcepoint

   Password: forcepoint

Figure 4. Jenkins Continuous Delivery Server Login Page

3. Deploy the Forcepoint Behavioral Analytics Stack from Forcepoint Continuous Delivery Server (Figure 5).

Figure 5. Forcepoint Continuous Delivery Server Dashboard

4. Optional: Check the deployment status from Forcepoint Continuous Delivery Server.
   a. The status and currently running deployment jobs can be found in the BuildExecutor Status window (Figure 6).

Figure 6. BuildExecutor Status Window

Create the Default UI Admin User.

1. Create the first admin user for the UI.

   Username: redowl@redowl.com

   Password: redowl

Important

By default, FBA does not ship with an initial user configured. The user must be manually created in order to login to the UI. The following commands must be executed on the postgres host from the command line.

Note

Do not copy and paste the text below, the line wrapping does not allow the commands to be executed correctly. These commands can be copied from the Jenkins container in `/usr/share/ro-ansible/sysconfdir/scripts/psql_admin_setup.sh`

```
psql -U redowlpostgres -d the_ui -c "INSERT INTO USERS (email, encrypted_password,
name, created_at, updated_at, password_updated_at) VALUES
('redowl@redowl.com','\$2a\$06\$mMhM9IWYk1J3Q15tGgP5rOryw7Mo1m3JL0eydVOtJ20gmm4twDKMW'
,'Red Owl', CURRENT_DATE, CURRENT_DATE, CURRENT_DATE);"
```

```
psql -U redowlpostgres -d the_ui -c "INSERT INTO roles_users (role_id, user_id)
(SELECT r.id, u.id FROM roles r INNER JOIN users u ON (u.email LIKE
'redowl@redowl.com') WHERE r.id != 13);"
```

```
psql -U redowlpostgres -d the_ui -c "INSERT INTO groups_users (group_id, user_id)
values (1,1);"
```

Appendix

DOCKER ROOTLESS

One of the new options introduced in the v3.4x architecture is the option to configure the solution for non-DDP customers using rootless docker. Rootless mode allows running the Docker daemon and containers as a non-root user to mitigate potential vulnerabilities in the daemon and the container runtime. Rootless mode does not require root privileges even during the installation of the Docker daemon.

This option has been made available for specific client use cases with extremely strict regulatory requirements. It is not recommended to use this installation path without explicit cause (i.e. regulatory compliance).

Advantages:

- Enable ability to mitigate potential vulnerabilities in the docker daemon.
- Allows installation of docker as a custom user.
- Allows a limited list of privileged commands.

Disadvantages:

- Kafka will not perform in any way in a rootless environment, and any connections requiring Kafka (such as DLP) will effectively be disabled.
- "One way" installation (i.e. once you have rootless as the installation path, you cannot switch to a standard docker installation without going through fresh install).
    - rootfull to rootless migration is not supported
- A large amount of additional effort is required to go through the installation process.

ROOTLESS ALLOW-LIST SUDO COMMANDS - ENHANCED PRIVILEGES ALLOW LIST

The installation process for the rootless FBA 3.4.2 release requires a small subset of commands that can be run with `sudo` during the installation.

The variables listed in the table below must be substituted for their actual values in the customer environment:

Table 1.3. Installation Variables

Varilable	Use/Definition	Example Value
`FBA_USER`	The username of the user FBA will be running as	centos
`FBA_UID`	The userid number of the `FBA_USER`	1000
`FBA_GROUP`	The group of the `FBA_USER`	centos
`FBA_DIR`	The directory where the FBA software will be installed	`/home/centos`

The table below contains commands that are required to be run with `sudo` on all hosts in the installation. Some of the commands are only required on specific hosts.

Table 1.4. Installation Commands

Command	Purpose	Hosts	Notes
`chown ${FBA_USER}:${FBA_GROUP} ${FBA_DIR}`	Ensure ownership of the `FBA_DIR` is correct.	All	Not needed if the ownership and group of the directory are already correct.
`cp ${FBA_DIR}/hosts /etc/hosts`	Include list of FBA hosts in `/etc/hosts`	All	Not needed if DNS is correctly set up and working on the host.
`loginctl enable-linger ${FBA_USER}`	Allows the rootless docker to continue running after the `FBA_USER` logs out.	All	N/A
`mkdir -p ${FBA_DIR}`	Ensure that the `FBA_DIR` is present.	All	Not needed if the directory is already present.

`rpm -Uvh ${FBA_DIR}/slirp4netns-0.4.3-4.el7_8.x86_64.rpm`
 	
`rpm -Uvh ${FBA_DIR}/yum-plugin-versionlock-1.1.31-54.el7_8.noarch.rpm`
	Install required system-level software.	All	`slirp4netns` improves network performance of port forwards in docker.
			`yum-plugin-versionlock` is used to ensure that `slirp4netns` is not updated during normal system upgrades so we can make sure the version installed has been tested with our environment.

`yum versionlock slirp4netns\*`	Version lock the `slirp4netns` rpm.	All	Note that this is not a wildcard, but rather a literal "*" character.

Command	Purpose	Hosts	Notes
`sed -i '/Service/a LimitMEMLOCK=infinity:infinity' /usr/lib/systemd/system/docker_${FBA_USER}.service`	Modify the rootless docker unit file in place to set the MEMLOCK limit to unlimited during docker startup.	es jenkins mds mon nifi postfix postgres rabbit redis rose ui ups	N/A
`setcap cap_net_bind_service=ep ${FBA_DIR}/bin/rootlesskit >/dev/null`	Allows the rootless docker executable to bind IP ports < 1024.	All	N/A

`sysctl --system >/dev/null`
`sysctl -w vm.max_map_count=262144`
`sysctl vm.overcommit_memory=1`
	Updates the `sysctl` parameters as required during the installation.	All	`vm.max_map_count` is required on some systems that user mmap on a large number of files (specifically for elastic search nodes and the monitoring node that also runs elastic). `vm.overcommit_memory` is used on some systems that allocate a large amount of virtual memory even if it is not going to be used (specifically for rabbit and redis hosts).

`systemctl daemon-reload`
`systemctl enable docker_${FBA_USER}.service`
`systemctl restart docker_${FBA_USER}`
`systemctl start docker_${FBA_USER}.service`
	Start, restart, enable the rootless docker systemd unit.	All	N/A

Several system files need to be written to during the installation process, this is done using the `tee -a filename` commands. Refer to the following table for the commands and the files they will affect.

Table 1.5. Files Edited at Installation

File	Use	Hosts	Text written to the File
`/etc/rc.local`	Disables `transparent_hugepages` at system boot time.	postgres redis	`echo never | tee /sys/kernel/mm/transparent_hugepage/enable`
`/etc/subgid`	Initializes the `gid` mapping utilized by the rootless docker system.	All	`${FBA_USER}:100000:65536`

File	Use	Hosts	Text written to the File
/etc/subuid	Initializes the uid mapping utilized by the rootless docker system.	All	${FBA_USER}:100000:65536
/etc/sysctl.d/01-max_user_namespaces.conf	Updates the max user namespaces at system boot time, required by rootless docker.	All	user.max_user_namespaces=28633
/etc/sysctl.d/01-overcommit-memory.conf	Updates the overcommit memory setting as described above at system boot time.	All	vm.overcommit_memory=1
/usr/lib/systemd/system/docker_${FBA_USER}.service	The rootless docker systemd unit file.	All	See the template below
/etc/sysctl.d/01-max-map-count.conf	Update the max_map_count vm setting for mmapped files as described above.	All	vm.max_map_count=262144
/sys/kernel/mm/transparent_hugepage/enabled	Modifies the transparent hugepage setting on the system as described above (in the section on the /etc/rc.local file), it is done here during the installation process so a reboot is not required.	postgres redis	Never

Template for the rootless docker systemd unit file

```
[Unit]
Description=Run dockerd rootless as user ${FBA_USER}
DefaultDependencies=no
After=network.target
[Service]
LimitNOFILE=65536:65536
Type=simple
User=${FBA_USER}
```

```
Group=${FBA_GROUP}
Environment="PATH=${FBA_DIR}/bin:/bin:/usr/bin:/sbin:/usr/sbin"
Environment="DOCKER_HOST=unix:///run/user/${FBA_UID}/docker.sock"
Environment="XDG_RUNTIME_DIR=/run/user/${FBA_UID}"
ExecStart=${FBA_DIR}/bin/dockerd-rootless.sh --experimental --storage-driver vfs
TimeoutStartSec=0
[Install]
WantedBy=default.target
```

The jenkins host will require the following software packages to be installed. The installation process performs this installation with a `rpm -Uvh --force` command. The rpms listed here are included in our `offline_installer` bundle.

```
bzip2-1.0.6-13.el7.x86_64.rpm
libseccomp-2.3.1-4.el7.x86_64.rpm
groff-base-1.22.2-8.el7.x86_64.rpm
perl-5.16.3-297.el7.x86_64.rpm
perl-Carp-1.26-244.el7.noarch.rpm
perl-Encode-2.51-7.el7.x86_64.rpm
perl-Exporter-5.68-3.el7.noarch.rpm
perl-File-Path-2.09-2.el7.noarch.rpm
perl-File-Temp-0.23.01-3.el7.noarch.rpm
perl-Filter-1.49-3.el7.x86_64.rpm
perl-Getopt-Long-2.40-3.el7.noarch.rpm
perl-HTTP-Tiny-0.033-3.el7.noarch.rpm
perl-PathTools-3.40-5.el7.x86_64.rpm
perl-Pod-Escapes-1.04-297.el7.noarch.rpm
perl-Pod-Perldoc-3.20-4.el7.noarch.rpm
perl-Pod-Simple-3.28-4.el7.noarch.rpm
perl-Pod-Usage-1.63-3.el7.noarch.rpm
perl-Scalar-List-Utils-1.27-248.el7.x86_64.rpm
perl-Socket-2.010-5.el7.x86_64.rpm
perl-Storable-2.45-3.el7.x86_64.rpm
perl-Text-ParseWords-3.29-4.el7.noarch.rpm
perl-Time-HiRes-1.9725-3.el7.x86_64.rpm
perl-Time-Local-1.2300-2.el7.noarch.rpm
perl-constant-1.27-2.el7.noarch.rpm
```

```
perl-libs-5.16.3-297.el7.x86_64.rpm
perl-macros-5.16.3-297.el7.x86_64.rpm
perl-parent-0.225-244.el7.noarch.rpm
perl-podlators-2.5.1-3.el7.noarch.rpm
perl-threads-1.87-4.el7.x86_64.rpm
perl-threads-shared-1.43-6.el7.x86_64.rpm
wget-1.14-18.el7_6.1.x86_64.rpm

parallel-20160222-1.el7.noarch.rpm
```

NOTES ON OPENVPN

The process of deploying OpenVPN should only be performed by Professional Serivces due to it's evolving customer deployment models.

Things to Consider

- The VPN host must be provisioned beforehand.
  - The VPN host must have SSH enabled and reachable from the provisioning ansible host.
  - The install and configuration are Ansible based.
    - `/etc/ansible/hosts` file must be accurate.
    - `/etc/ansible/group_vars/all` must be accurate and tailored to any site-specific overrides necessary.
    - Host machine running ansible playbooks must have ssh access to all hosts in the `/etc/ansible/hosts` inventory file.

DEPLOYING OPENVPN

1. Create the Baseline FBA VPN host:

```
ansible-playbook ro-baseline.yml --limit openvpn
```

   a. Ensure that the SSH key is copied to the FBA VPN host by running the following command:

```
cp user.pem ~/.ssh/user.pem
chmod 600 ~/.ssh/user.pem
```

   b. Retrieve the FBA VPN host Public IP by running the following command:

```
curl ipecho.net/plain
```

   c. Install common FBA packages.

      i. Option 1: Run everything by running the following command:

```
ansible-playbook ro-common.yml --limit openvpn
```

      ii. Option 2: Run select playbooks, based on customer needs:

         i. Always run

```
ansible-playbook common.yml
```

> **Note**
>
> This should not be confused with `ro-common.yml`

         ii. Optionally run

```
ansible-playbook selinux.yml --limit openvpn
ansible-playbook ntp.yml --limit openvpn
ansible-playbook hostname.yml --limit openvpn
ansible-playbook ro-ssh.yml --limit openvpn
ansible-playbook hosts_file.yml --limit openvpn
```

d. Deploy the OpenVPN Service.

```
ansible-playbook openvpn.yml
```

e. Start the OpenVPN Service.

i. Run from the FBA VPN host:

```
sudo systemctl restart openvpn@server.service
```

f. Create the OpenVPN Users:

Note
Substitute `{{user}}` with correct username.

i. Run from the FBA VPN host:

```
sudo /etc/openvpn/addvpnuser.sh fp-ueba-ops-{{user}}
sudo su - {{user}}
passwd - enter password twice when prompted
cp /etc/openvpn/keys/{{user}}-vpn-*.tar.gz /home/{{user}}
```

ii. Copy `/home/{{user}}-vpn-*.tar.gz` to remote machine for Professional Services Engineer use.

g. Configure 2FA - Google Authenticator.

Note
Substitute `{{user}}` with correct username.

i. Run from FBA VPN host logged in as the newly created user:

i. google-authenticator

i. The correct question answers are: Y, Y, Y, N, Y.

ii. Copy the barcode and/or the url to add to the authenticator app.

h. Test FBA VPN connection.

Note
Substitute `{{user}}` with correct username.

i. Run from Professional Services OSX host:

```
tar {{user}}-vpn-*.tar.gz -C {{user}}-vpn.tblk
```

ii. Drag and drop `{{user}}-vpn.tlbk` into tunnelblick configuration windows.

iii. Connect using username,password+googleauth.

TROUBLESHOOTING OPENVPN

- If authentication fails, ensure the password is set correctly. Reset password as necessary.
  - Google-authenticator may need to be rerun.
- If name lookups are failing there is a bug in the tunnelblik software to where the client does not push the AWS DNS server and search domains to the local machine.
  - This can be fixed by manually adding the route53 address x.x.x.2 for the DNS server and appropriate search domain using the primary network interface.

AWS ENCRYPTION OPTIONS FOR NATIVE AND ATTACHMENT STORAGE

FBA supports various means of encryption options in AWS S3 for Native and Attachment storage in the Conversion Service. The default used is SSE-S3. Alternatively, SSE-C or SS3-KMS can be enabled. No UI configuration changes are necessary to enable either SSE-C or SSE-KMS, but the AWS IAM credentials used by the UI must be on the KMS key policy.

To enable one of the alternative AWS encryption options, alterations must be made to:

```
/usr/share/ro-ansible/roles/ro-conv/defaults/main.yml
```

Default Values:

```
# encryption for S3 storage; supported types are (sse-s3, sse-c, ss3-kms)
natives_encryption_type: sse-s3
attachments_encryption_type: sse-s3
# required if sse-c is enabled
natives_sse_c_key_file: ""
attachments_sse_c_key_file: ""
# required if sse-kms is enabled
natives_sse_kms_key_arn: ""
attachments_sse_kms_key_arn: ""
```

To enable sse-c:

```
# encryption for S3 storage; supported types are (sse-s3, sse-c, ss3-kms)
natives_encryption_type: sse-c
attachments_encryption_type: sse-c
# required if sse-c is enabled

natives_sse_c_key_file: "/path/to/my.key"
attachments_sse_c_key_file: "/path/to/my.key"
# required if sse-kms is enabled
natives_sse_kms_key_arn: ""
attachments_sse_kms_key_arn: ""
```

To enable ss3-kms:

```
# encryption for S3 storage; supported types are (sse-s3, sse-c, ss3-kms)
natives_encryption_type: ss3-kms
```

```
attachments_encryption_type: ss3-kms
# required if sse-c is enabled
natives_sse_c_key_file: ""
attachments_sse_c_key_file: ""
# required if sse-kms is enabled
natives_sse_kms_key_arn:
"arn:aws:kms:<region>:<account>:key/<key>"
attachments_sse_kms_key_arn:
"arn:aws:kms:<region>:<account>:key/<key>"
```

MANUALLY RUN ANSIBLE PLAYBOOKS

Prepare Forcepoint Behavioral Analytics Stack

1. FBA host names:

```
ansible-playbook hostname.yml
ansible-playbook hosts_file.yml
```

   a. FBA baseline:

```
ansible-playbook ro-baseline.yml
```

   b. Install common FBA packages.

      i. Option 1: Run everything.

```
ansible-playbook ro-common.yml
```

      ii. Option 2: Run select playbooks, based on customer needs.

         i. Always run:

Note
This should not be confused with `ro-common.yml`.

```
ansible-playbook common.yml
```

         ii. Optionally run:

```
ansible-playbook selinux.yml
ansible-playbook ntp.yml
ansible-playbook ansible-openssh.yml
```

   c. Deploy FBA secrets:

```
ansible-playbook vault.yml
```

   d. To deploy FBAs Middleware, deploy the Jenkins host:

```
ansible-playbook jenkins.yml
```

   e. Deploy Redis:

```
ansible-playbook redis.yml
```

   f. Deploy Postgressql:

```
ansible-playbook postgres.yml
```

   g. Deploy RabbitMQ:

```
ansible-playbook rabbit.yml
```

h. Deploy Kafka

```
ansible-playbook kafka.yml
```

i. Deploy ElasticSearch:

```
ansible-playbook ro-es.yml
```

j. Deploy Monitoring Elastic Search:

```
ansible-playbook ro-mon-es.yml
```

k. Initialize FBA Schema:

```
ansible-playbook ro-schema.yml
```

l. Deploy FBA Monitoring Software:

```
ansible-playbook ro-monitoring.yml
```

m. Deploy FBA Master Data Service:

```
ansible-playbook ro-mds.yml
```

n. Deploy FBA Master Data Service analytics node:

```
ansible-playbook ro-mdslytics.yml
```

o. Deploy FBA API Service:

```
ansible-playbook ro-api.yml
```

p. Deploy FBA Queue Worker Service:

```
ansible-playbook ro-qw.yml
```

q. Deploy FBA Conversion Service:

```
ansible-playbook ro-conv.yml
```

r. Deploy FBA Content Service:

```
ansible-playbook ro-cont.yml
```

s. Deploy FBA UPS Service:

```
ansible-playbook ro-ups.yml
```

t. Deploy Rose Service:

```
ansible-playbook ro-rose.yml
```

u. Deploy Apache Nifi Service:

```
ansible-playbook ro-nifi.yml
```

v. Deploy Forcepoint UI Service:

```
ansible-playbook ro-ui.yml
```

w. Deploy Logstash:

```
ansible-playbook ro-logstash.yml
```

x. Deploy Kibana:

```
ansible-playbook ro-kibana.yml
```

y. Deploy Forcepoint Integration Service (optional):

```
ansible-playbook ro-api.yml
```

z. Deploy Security Features (optional):

```
ansible-playbook ro-jobs.yml -i /etc/ansible/hosts -t
tls-version -f 5 -e set_tls_version=true -v
```

DEPLOYING CURATOR

The `deploy-ueba-curator` job was removed from the deploy stack process as it requires Jenkins to restart at the end of the job. This causes the deployment process to appear as though it failed. Manually run the `deploy-ueba-curator` job after the install process is complete.