Replicator Docs

Release 20.0.2

Plixer, LLC

Apr 20, 2026

VERSION 20.0.2

Changed in version 20.0.0: Replicator and Scrutinizer now share the same Plixer One Platform architecture, which enables a unified UI as well as shared core functions.

This manual covers topics specific to Replicator. For documentation related to **alarms**, **reporting**, **administrative controls**, and other shared functionality, refer to the Scrutinizer manual.

CHAPTER

ONE

GETTING STARTED

Virtual appliances

Deploy your ESXi, Hyper-V, KVM, Proxmox, or AWS virtual appliance

Virtual appliance deployment **Hardware appliance**

Deploy your hardware appliance

Hardware appliance deployment **Headless deployments**

Deploy additional “headless” instances for a standalone Replicator or Scrutinizer deployment

Headless deployments **Appliance setup**

Complete initial setup and licensing after deployment

Basic configuration

CHAPTER

TWO

USING REPLICATOR

Profiles

Set up replication profiles with exporter policies and collector assignments

Profiles **Collectors**

Add and manage destination hosts for replicated packet streams

Collectors **Exporters**

View and monitor UDP packet-exporting devices sending streams to Replicator

Exporters **Admin**

Access appliance monitoring and administrative functions

System management

CHAPTER

THREE

ADVANCED SERVICES

Replicator APIs

Leverage Replicator APIs for external integration

Replicator APIs **Auto Replicate**

Automatic stream management for multiple collectors

Auto Replicate **High availability**

Set up redundant Replicator instances for fault tolerance

High availability **Version upgrades**

Upgrade procedures and instructions

Version upgrades **Profile migration**

Migrate profile data to v20.0.0+ deployments

Profile migration

CHAPTER

FOUR

HELP AND REFERENCES

FAQ

Answers to frequently asked questions

FAQ **Changelog**

Version history and release notes

Replicator changelogs **Glossary**

Glossary of terms used in Replicator

Glossary **Attributions**

Open source and third-party licenses

Third-party attributions About Replicator

Replicator is a networking appliance that can collect, replicate, and load-balance streams from UDP metadata-exporting devices before forwarding them to any number of destination hosts. Replication functions are managed through user-defined profiles, eliminating the typical 1:1 pairing requirement between flow exporting devices and network intelligence collectors.

- **Fully configurable replication** - Create profiles to manage exporter-to-collector assignments and use inclusion/exclusion policies to keep profiles updated as your environment expands
- **Insightful, real-time alarms** - Gain additional visibility into all devices sending flows to the Replicator and get alerted to drops in traffic and other irregularities
- **Full-featured web interface** - View and manage the Replicator environment in an intuitive web interface
- **Robust command line interface (CLI)** - Use interactive commands to configure Replicator's functions and leverage advanced features

For further questions, check out the *FAQ page* or contact *Plixer Technical Support*.

4.1 Deployment Guides

Replicator virtual appliances can be deployed in local hypervisors, Amazon Web Services (as an AMI via the AWS Marketplace), Google Cloud Platform, Microsoft Azure, or Oracle Cloud Infrastructure. *Hardware appliances* are also available upon request.

Contact *Plixer Technical Support* or a local reseller for availability and licensing or visit www.plixer.com to learn more.

Note

Profile data from a Replicator 19.1.1 appliance can be migrated to a different v20.0.2 instance. Refer to *this guide* for further details.

On this page:

Note

- The information in this section applies specifically to Replicator 20.0.2 standalone deployments and "headless" appliances. Refer to the Scrutinizer manual for instructions to enable the local Replicator instance on a Scrutinizer deployment. Contact *Plixer Technical Support* to learn more about licensing options.
- Standalone Replicator deployments can be upgraded by adding a Plixer One/Scrutinizer license under **Admin > Plixer > Scrutinizer** in the web interface. However, this will require the instance to be provisioned with additional resources as described in these Scrutinizer deployment guides and sizing recommendations.

4.1.1 Virtual appliance deployment

Basic requirements for virtual appliances:

Component	Recommended (for production environments)
Memory	8 GB
Storage	100 GB 15K RAID 0 or 10 configuration
Processor	2 CPU cores, 2.0+ GHz

Note

A single Replicator instance can replicate packets at rates close to line speed with sufficient CPU provisioning. However, the interface configuration and number of destination collectors must also be taken into consideration. If all packets are being replicated to two destinations, the outbound bandwidth utilization will be twice the inbound volume.

4.1.1.1 Local hypervisors

ESXi deployment

Additional requirements:

- ESXi 6.7 U2+
- VMware vSphere or vCenter

Deploying the OVF template

1. Log in to the Plixer Customer Portal or use the link provided by *Plixer Technical Support* to download the latest VMware virtual appliance package.
2. Extract the contents of the package to a location on the ESXi server.

3. In vSphere or vCenter, deploy the appliance on a host using the *OVF template* option (this will require the OVF and VMDK files).
4. Select *Thick Provision* for the datastore disk format.
5. After selecting the network to be used by the virtual appliance, verify the configuration in the summary before starting the import operation.
6. After the template has been successfully imported (may take several minutes), assign a static MAC address to the Replicator NIC for licensing purposes.
7. Power on the VM.

After the Replicator virtual appliance completes booting, proceed with the *initial appliance setup*.

> **Note**
>
> To upgrade the virtual machine's hardware version to the latest ESXi version, select **Compatibility > Upgrade VM Compatibility** in vSphere or vCenter while the VM is powered off. When the VM is powered back on after the upgrade, it will boot up with the latest ESXi hardware version available.

Hyper-V deployment

Additional requirements:

- Generation 2 Hyper-V VM
- Hyper-V 2012
- Hyper-V Manager

Deploying the Hyper-V virtual appliance

1. Log in to the Plixer Customer Portal or use the link provided by *Plixer Technical Support* to download the latest Hyper-V virtual appliance package.
2. Extract the contents of the package to a location on the Hyper-V server.
3. In Hyper-V Manager, select the option to import a VM, and then select the **Replicator Hyper-V** image.
4. After the image has been imported, provision the Replicator VM based on the *recommended resources*.
5. Select a network adapter and assign it to the appropriate virtual switch.
6. Assign a static MAC address to the VM.
7. Save the updated settings, and then start the VM.

After the Replicator virtual appliance completes booting, connect to the VM and then proceed with the *initial appliance setup*.

KVM deployment

Additional requirements:

- KVM 16 or higher

Deploying the KVM virtual appliance

1. Log in to the Plixer Customer Portal or use the link provided by *Plixer Technical Support* to download the latest KVM virtual appliance package.
2. Create a directory for the install:

```
mkdir /kvm/replicator_vm/
```

3. Extract the contents of the package to the new directory:

```
sudo tar xvzf PACKAGE_FILENAME.tar.gz -C /kvm/replicator_vm/
```

4. Run the installation script in the new directory:

```
cd /kvm/replicator_vm/PACKAGE_FILENAME
sudo ./install-kvm-scrut.sh
```

5. Wait for the confirmation that the virtual machine has been created from the image.

After the Replicator virtual appliance completes booting, access the console using `virsh console <VM_DOMAIN_OR_ID>` to proceed with the *initial appliance setup*.

Nutanix

Deploying the virtual appliance in Nutanix

1. Log in to the Plixer Customer Portal or use the link provided by *Plixer Technical Support* to download the latest qcow2 image file:
2. Log in to Prism Element and upload the image (as a disk) to any storage container (except *SelfServiceContainer*).
3. After the image becomes active, create a new VM with the following configuration:
   - *Resources*: *Recommended resources* (minimum of 8 cores and 16 GB RAM, fewer CPUs with more cores is recommended)
   - *Boot configuration*: UEFI
   - *Operation*: Clone from image
   - *Bus type*: SATA (SCSI is not recommended due to known issues with Red Hat 9 systems)
   - *Image*: Image/disk uploaded in step 3
   - *Index*: Next available
4. Add a new NIC to the VM and assign it to the desired subnet.
5. Save the VM configuration, and then power on the VM.

After the Replicator virtual appliance completes booting, launch the console to proceed with the *initial appliance setup*.

Proxmox deployment

To deploy the Replicator virtual appliance in Proxmox, follow these steps:

> **Note**
>
> - When attaching the imported disk (step 6), verify that its name matches what's displayed in the GUI.

- The syntax in the instructions below should be modified to match the actual VMID and disk names/numbers used.

Deploying the virtual appliance in Proxmox

1. Log in to the Plixer Customer Portal or use the link provided by *Plixer Technical Support* to download the latest qcow2 image file.
2. Create a new virtual machine in Proxmox with the following configuration:
   - BIOS: OVMF (UEFI)
   - SCSI controller: VMware PVSCSI
   - Network adapter: E1000
   - CPU/memory: *Recommended resources*
   - Add a new EFI disk with default sizing
3. Import the disk via the CLI:

   ```
   qm importdisk VMID /var/lib/vz/template/Plixer_Replicator.qcow2 ZFS_DISK_NAME
   ```

   Example:

   ```
   qm importdisk 100 /var/lib/vz/template/Plixer_Replicator.qcow2 local-zfs
   ```

4. Attach the imported disk to the virtual machine:

   ```
   qm set VMID -scsi0 local-zfs:VM_DISK_NAME
   ```

   Example:

   ```
   qm set 100 -scsi0 local-zfs:vm-101-disk-1
   ```

5. Remove/delete the unused disk (the default disk created when the VM was added in Proxmox).
6. Start the VM.

After the Replicator virtual appliance completes booting, access the console to proceed with the *initial appliance setup*.

4.1.1.2 Cloud platforms

AWS AMI deployment

Deploying the Scrutinizer AMI

After subscribing to the service via the AWS Marketplace product page, deploy the Replicator AMI by creating/launching a new EC2 instance with the following configuration:

- *Names and tags*: Configure the name, resource types, and optional tags for the instance.
- *Application and OS images*: Select the Replicator AMI from the **My AMIs** tab.
- *Instance type*: Select *C5.2xlarge* for flow rates up to 10,000 flows per second (contact *Plixer Technical Support* for assistance if the expected flow volume exceeds that).
- *Key pair*: Select or create a new key pair to assign to the instance.
- *Network settings*: Select the VPC, subnet, and security group to assign the instance to.

> **Important**
>
> Because an active instance's primary private IP address cannot be released, we recommend deploying the AMI with two NICs and using the secondary as the collection interface.

- *Storage*: Leave the size of the root volume (`/dev/xvda/`) at the default 100 GB.
- *Advanced details*: Set *Shutdown behavior* to **Stop** and *Termination protection* to **Enabled**.

After the instance has been launched, access the Replicator web interface via the instance's primary private or public IP address, and then proceed to *add a license*.

> **Note**
>
> - For AMI deployments, the default password for the web interface `admin` user is the AWS instance ID of the Replicator instance, which can be copied from the **Instance Summary** view of the EC2 interface.
> - Use the following command to SSH to the server as the `plixer` user after the instance has been launched:
>
> ```
> ssh -i PATH_TO_KEY/key.pem plixer@REPLICATOR_IP
> ```

Google Cloud Platform

Additional requirements:

- A GCP project with *Billing*, *Compute Engine*, and *Migrate to Virtual Machines* enabled
- Permissions to create *Compute Engine images*, *Compute Engine VM instances*, and *Cloud Storage buckets* (if not using an existing bucket)
- A cloud storage bucket on the region intended for the VM (for staging the image)

Importing and deploying the Replicator VM

1. Log in to the Plixer Customer Portal or use the link provided by *Plixer Technical Support* to download the latest VMware virtual appliance OVA package.
2. Upload the image to the staging bucket.
3. Select the option to import a **machine image** and use the following settings:
   - *Source*: Cloud Storage
   - *File*: Select the uploaded OVA
   - *Operating system*: RHEL 9

   This operation will create a reusable custom image and may take up to 15 minutes. The image must be successfully imported before the Scrutinizer VM can be created.
4. Create a new VM instance with the machine type most closely matching the *recommended resources* for the expected flow volume (*n4* or *c4* recommended).
5. Configure the OS and storage settings for the VM as follows:
   - *Boot disk*: The imported Replicator image
   - *Disk type*: *Hyperdisk Balanced* (required for C4/N4 machine types)
   - *Disk size*: Adjust to match storage requirements
6. Configure the networking settings for the VM as follows

- Assign an external IPv4 address (ephemeral).
- Enable HTTPS traffic through the firewall.
- Add a network tag: *replicator-https*.
- Assign a hostname (optional but recommended).

7. Verify that all settings were configured correctly, and then create/launch the VM.

After the instance has been launched, connect to the VM via serial console (see below if not already enabled for the project) to proceed with the *initial appliance setup*.

Enabling serial console access

Serial console access (project-level setting) can be enabled for first boot validation and troubleshooting.

In the GCP console, edit the metadata settings for the Compute Engine to add the following:

- *Key*: `serial-port-enable`
- *Value*: `true`

The option to connect to the Replicator VM via serial console will become available after the new key is saved.

Microsoft Azure

Additional requirements:

- A Windows 10+ or Windows Server host with Internet access, at least 200 GB free disk space, and Hyper-V installed
- Administrator permissions (including PowerShell commands) on the Windows host
- Administrator credentials for the Azure account the Replicator virtual appliance will be deployed on

Uploading and deploying the Replicator VM

Important

Replace the file paths in step 3 and 6 below with the correct paths to the downloaded and converted files in your environment.

1. Log in to the Plixer Customer Portal or use the link provided by *Plixer Technical Support* to download the latest Hyper-V virtual appliance package on the Windows host.
2. Extract the VHD (`Replicator.vhdx`) from the file.
3. Start a PowerShell session on the Windows host, and then convert the disk image to fixed size in Powershell:

```
Convert-VHD -Path 'C:\Users\User Name\Downloads\Replicator-hyperv\Replicator_Hyper-
↪V\Virtual Hard Disks\Replicator.vhdx' -Destination 'C:\tmp\Replicator.vhd' -
↪VHDType Fixed
```

4. Install the **Az PowerShell module**:

```
Install-Module -Name Az
```

5. Authenticate the Windows PowerShell session with the Azure account to be used for deployment:

```
Connect-AzAccount
```

> **Note**
>
> If the connection fails after the correct Azure credentials are entered, run the following:
>
> ```
> Set-ExecutionPolicy RemoteSigned
> ```

6. Upload the Replicator VHD to Azure as a managed disk (replace `RESOURCE_GROUP`, `AZURE_REGION`, and `DISK_NAME` below with the correct details):

```
Add-AzVhd -LocalFilePath 'C:\tmp\Replicator.vhd' -ResourceGroupName RESOURCE_GROUP -
↪Location AZURE_REGION -DiskName DISK_NAME -DiskHyperVGeneration V2 -DiskOsType␣
↪Linux
```

7. After the Replicator VHD has been uploaded, deploy a new VM using the disk image from the Azure portal (note the IP address assigned to the VM as this will be required when setting up the appliance).
8. Launch/start the VM.

After the Replicator VM completes booting, SSH to the IP address assigned as the `plixer` user to proceed with the *initial appliance setup*.

Oracle Cloud Infrastructure

Additional requirements:

- A cloud storage bucket (for staging the image)
- Gateway and netmask of the OCI VNC subnet that Replicator will be deployed on

Importing and deploying the Replicator VM

1. Log in to the Plixer Customer Portal or use the link provided by *Plixer Technical Support* to download the latest VMware virtual appliance package.
2. If necessary, extract the OVA (`Replicator_Vmware_20.0.2-bios.ova`) from the file.
3. Upload the image to the storage bucket.
4. Create a new custom image by importing the uploaded file from the storage bucket with the following settings:
   - *Operating system*: Oracle Linux
   - *Image type*: VMDK
   - *Launch mode*: Emulated (required)
5. Create a new VM instance using the custom image and configure the following settings:
   - Select the custom image created in the previous step.
   - Select an image shape (e.g., VM.Standard.E5.Flex) and expand the CPU core count and memory allocation to match the *recommended resources*.
   - Enter a primary VNIC name (required for the Replicator VM).
   - Manually assign a private IPv4 address to use as the static address for the Replicator appliance (must be entered during appliance setup).
   - Add public or generated keys for SSH access.

- Adjust the boot volume size based on *these storage recommendations* and keep VPU at the default value.

6. Save the instance configuration and start/launch the VM.

After obtaining the required details, SSH to the VM as the `plixer` user to proceed with the *initial appliance setup*.

4.1.2 Hardware appliance deployment

Replicator hardware appliances support higher flow rates due to their dedicated resources and are strongly recommended for environments with extremely high flow replication requirements. They are available through *Plixer Technical Support*.

After receiving your package, follow *this guide* to deploy the appliance.

Hardware setup

After removing the Replicator hardware appliance from its packaging, verify that all accompanying accessories (rack-mount kit, appliance-locking bezel and keys, and power cord) are included. The appliance can be mounted in a standard 19-inch rack or cabinet.

Important

If your box arrives torn, dented, or otherwise damaged, the appliance itself seems damaged, or there are missing parts, contact *Plixer Technical Support* immediately and **do not attempt to install the unit**.

From there, follow these steps to set up the Replicator hardware appliance:

1. Connect the appliance to the network as indicated by the port labels on the rear panel.
2. Connect the power cable to one of the power supply sockets and plug the other end to a grounded AC outlet or UPS (if the appliance has redundant PSUs, connect each socket to an independent power source).
3. [Optional] Connect the iDRAC port to a remote access controller using an RJ-45 cable to enable remote console access for hardware management and monitoring. Contact *Plixer Technical Support* for help with configuring alerts for hardware-related events.
4. Using the additional ports provided, connect a monitor and keyboard to use during the appliance's initial setup.

Once the Replicator hardware appliance has been set up and cabled, power it on and proceed with the *initial appliance setup*.

4.1.3 Headless deployments

Additional Replicator instances (for greater replication capacity, *high availability*, etc.) can be deployed as "headless" instances to minimize their resource footprint. These deployments do not include the web interface component; they must be registered and managed from a remote administrative/configuration Replicator instance (either a standalone Replicator instance or a Plixer One/Scrutinizer environment) and cannot be configured independently.

Note

- Hardware appliances that are upgraded to v20.0.2 can be used as either standalone (default) or headless instances. See *these instructions* for more details.
- Additional Replicator instances must be supported by the current license key. Contact *Plixer Technical Support* for further details.

4.1.3.1 Registering a headless Replicator instance

Before deploying a headless Replicator instance, it must first be registered on the admin instance as follows:

View instructions

1. Navigate to **Admin** > **Resources** > **Replicators** in the web interface.
2. Click the **Add** button.
3. Enter a name to assign to the new Replicator instance, and then click **Save**.
4. Click on the name of the new instance in the main view and note the authentication token shown in the tray.

Once the new instance has been registered, proceed to deploying the headless Replicator instance.

4.1.3.2 Deploying a headless Replicator instance

Follow these steps to deploy a headless Replicator instance after it has been registered:

View instructions

1. Download the latest headless Replicator VM package for your hypervisor from the Plixer Customer Portal.
2. Deploy the VM following the instructions *here*.
3. Complete the *basic appliance configuration*.
4. After it reboots, SSH to the appliance as the `plixer` user again.
5. Enter the following details when prompted:
   - IP address of the admin/configuration instance (typically the IP address of the standalone Replicator instance or a Scrutinizer primary reporter)
   - Authentication token generated when the headless instance was registered (see above)
   - Name given to the headless instance when it was registered
6. Return to the Admin > Resources > Replicators page in the admin instance web interface and verify that the new instance has successfully self-registered with the correct IP address.

Once the headless Replicator instance has been successfully registered and deployed, it can be configured for standalone replication or used in a *high-availability pair*.

4.1.3.3 Changing IP addresses (headless instances only)

Follow these steps to change the IP address of a headless Replicator instance:

1. SSH to the instance as the `plixer` user:

   ```
   ssh plixer@HEADLESS_IP_ADDRESS
   ```

2. Stop the Replicator service:

   ```
   sudo systemctl stop replicator
   ```

3. Update `IPADDR` in `/etc/sysconfig/network-scripts/ifcfg-eth0` (or `ifcfg-bond0`) with the new IP address (`IP_NEW`):

   ```
   sudo sed -i 's/^IPADDR=.*/IPADDR=IP_NEW/' /etc/sysconfig/network-scripts/ifcfg-eth0
   ```

   or

```
sudo sed -i 's/^IPADDR=.*/IPADDR=IP_NEW/' /etc/sysconfig/network-scripts/ifcfg-bond0
```

4. On the admin instance (standalone Replicator or Scrutinizer instance), run the following queries to replace the previous IP address (`IP_PREV`) with the new IP address (`IP_NEW`) in the configuration:

```
psql plixer

BEGIN;
ALTER TABLE replicator.profiles DROP CONSTRAINT profiles_replicator_ip_fkey;
ALTER TABLE replicator.profiles ADD constraint profiles_replicator_ip_fkey FOREIGN␣
↪KEY(replicator_ip) REFERENCES replicator.deployments(replicator_ip) ON UPDATE␣
↪CASCADE ON DELETE CASCADE;

DELETE FROM replicator.collector_state  where replicator_ip='IP_PREV';
DELETE FROM replicator.exporter_state  where replicator_ip='IP_PREV';

UPDATE replicator.deployments SET replicator_ip='IP_NEW' WHERE replicator_ip='IP_
↪PREV';
UPDATE replicator.deployments SET paired_ip='IP_NEW' WHERE paired_ip='IP_PREV';
COMMIT;
```

5. Restart the Replicator service:

```
sudo systemctl start replicator
```

When done, navigate to **Admin** > **Resources** > **Replicators** to verify that the IP address of the headless Replicator has been updated successfully.

4.1.4 Basic configuration

After deploying and starting the appliance, follow the basic configuration steps below to prepare Replicator for use.

4.1.4.1 Initial setup

After the Replicator appliance completes its first boot sequence, log in with the credentials `plixer:plixer` to start the initial setup script:

1. Provide the following information when prompted by the script:
   - Static IP address
   - Netmask
   - Gateway
   - FQDN
   - DNS IP address
   - NTP server IP address
2. Enter any additional information requested.
3. At the end of the script, press *Enter* to apply the settings and wait for the server to reboot again to apply the settings.

After the final appliance reboot, log in to the web interface at the IP address provided with the default `admin:admin` credentials and proceed to *add a license*.

Note

- The default password for the web interface `admin` account can be changed from the Admin > Users & Groups > User Accounts page.
- The default self-signed certificate can be *replaced with a CA-signed certificate* if desired.

4.1.4.2 Adding a license

To add/register a Replicator license key, navigate to **Admin > Plixer > Replicator Licensing** in the web interface after completing the *initial appliance setup process*.

A license key can be obtained by contacting *Plixer Technical Support* and providing them with the *Machine ID* displayed on the licensing page. The key should then be pasted into the *License Key field* and saved.

Details for the current license (validity, appliance/server counts, etc.) will be displayed on the page after a key has been added.

Note

- For AWS AMI deployments, the default password for the web interface `admin` user is the instance ID of the Replicator instance, which can be copied from the **Instance Summary** view of the AWS console.
- Replicator now shares the same UI with Plixer One/Scrutinizer (requires Scrutinizer 19.7.0+ and Replicator 20.0.0+). The IP address for accessing the web interface is assigned during the *initial setup process* after the appliance first boots.

4.1.4.3 Configuring SSL

SSL support is automatically enabled during the initial setup process for a standalone Replicator instance. A self-signed SSL certificate with default values is created at the same time.

This self-signed certificate can later be replaced with a CA-signed certificate if desired.

Installing a CA-signed SSL certificate

As long as the system is set to use the self-signed SSL certificate created during the initial setup process, browsers will return an untrusted certificate warning, which users must override to access the web interface.

To avoid this, an SSL certificate that has been signed by an internal or commercial Certificate Authority (CA) will need to be installed.

Generating a custom certificate signing request (CSR)

1. SSH to the primary reporter as the `plixer` user:

   ```
   ssh plixer@PRIMARY_REPORTER_IP
   ```

2. [Optional] Create a new directory for the custom CSR, keys, and certificates:

   ```
   sudo mkdir /home/plixer/CustomCerts
   ```

   This will provide a static location for storing and managing future certificates.

3. Create a CSR config/details file:

```
sudo touch /home/plixer/CustomCerts/csr_config.txt
```

Tip

- If the details for the CSR do not change from year to year, `csr_config.txt` can be re-used to create a new CSR when the old certificate expires.
- When generating a new CSR, key, and certificate, including a date in the filename will help identify the correct files in case future changes (e.g., upgrades) overwrite the existing certificate.

4. Add the details for the CSR to `csr_config.txt` in the following format:

```
[req]
default_bits=2048
prompt=no
default_md=sha256
req_extensions=req_ext
distinguished_name=dn

[dn]
C=US
ST=Maine
L=Kennebunk
O=Plixer, LLC
OU=IT
emailAddress=support@plixer.com
CN=replicator.plxr.local

[req_ext]
subjectAltName=@alt_names

[alt_names]
DNS.1=replicator.plxr.local
```

> **Note**
>
> `[alt_names]` is now required. To specify multiple Subject Alternative Names (SANs), use one line for each entry, with incrementing DNS numbers (DNS.2=, DNS.3=, etc.).

5. Generate the new CSR and key:

```
cd /home/plixer/CustomCerts
sudo openssl req -new -sha256 -nodes -out newRequest.csr -newkey rsa:4096 -keyout␣
↪newCaKey.key -config csr_config.txt
```

The custom CSR (`/home/plixer/CustomCerts/newRequest.csr`) can then be sent to any preferred CA for signing.

Installing the signed certificate

> **Important**
>
> In some cases, Replicator 19.01 and Scrutinizer 19.5.x deployments will also have `localhost.crt` and `localhost.key` files in addition to `ca.crt` and `ca.key`. These files were generated during the deployment/upgrade process but should not be used.
>
> The following steps will ensure that the correct certificates are in place and in use:
>
> #### View instructions
>
> 1. Verify `localhost.crt` and `localhost.key` do not exist on the appliance:
>
> ```
> sudo ls /etc/pki/tls/certs/
> sudo ls /etc/pki/tls/private/
> ```
>
> If neither file exists, no further action is required.
>
> 2. If either of the previous commands discovers the corresponding `localhost` file, update the appliance to look for the correct files:
>
> ```
> sudo sed -i 's/localhost.crt/ca.crt/g' /etc/httpd/conf.d/ssl.conf
> sudo sed -i 's/localhost.key/ca.key/g' /etc/httpd/conf.d/ssl.conf
> sudo chmod 600 /etc/pki/tls/certs/ca.crt
> sudo chmod 600 /etc/pki/tls/private/ca.key
> sudo mv /etc/pki/tls/certs/localhost.crt /etc/pki/tls/certs.ca.crt
> sudo mv /etc/pki/tls/private/localhost.crt /etc/pki/tls/private/ca.key
> ```
>
> 3. Restart the httpd service:
>
> ```
> sudo systemctl restart httpd
> ```

After receiving the CA-signed certificate, follow these steps to install it:

1. Copy the new certificate to the `/home/plixer/CustomCerts` directory (or any temporary directory if `CustomCerts` was not previously created) on the Replicator server.
2. Backup the current CA certificate and key:

```
sudo cp /etc/pki/tls/certs/ca.crt /etc/pki/tls/certs/ca.crt.bak
sudo cp /etc/pki/tls/private/ca.key /etc/pki/tls/private/ca.key.bak
```

3. Move the new certificate to the correct location:

```
cp /home/plixer/CustomCerts/CA_CERT_FILENAME.crt /etc/pki/tls/certs/ca.crt
```

4. Move the new key generated with the CSR to the correct location:

```
sudo cp /home/plixer/CustomCerts/NEW_KEY_FILENAME.key /etc/pki/tls/private/ca.key
```

   If the `CustomCerts` directory was not created/used, the key can be found in the same directory the CSR was generated in.

5) Restart the nginx service (httpd on pre-v20.0.0 Replicator or pre-v19.7.0 Scrutinizer deployments):

```
sudo systemctl restart nginx
```

To verify that the web interface is using the correct SSL certificate, use a browser to navigate to the login page using the FQDN specified in the CA-signed certificate. The browser should no longer return an untrusted certificate warning and the padlock icon in the address bar should be locked instead of open.

Note

The private key may need to be encrypted with the `/usr/bin/ask.sh` passphrase:

```
openssl rsa -in server.key -out server.key.new
```

Non-default CSR configurations

Certificate signing requests can also be generated with non-default configurations (stronger encryption, no email address, etc.) using the values in the `csr_config.txt` file in the *above instructions*.

After the desired configuration has been saved, continue to follow the same instructions to generate the CSR and install the CA-signed certificate.

4.2 Features and Functionality

4.2.1 Replicator UI

Exporters

View and monitor UDP packet-exporting devices sending streams to Replicator

Exporters **Profiles**

Set up replication profiles with exporter policies and collector assignments

Profiles **Collectors**

Add and manage destination hosts for replicated packet streams

Collectors **Overview**

Monitor system/appliance activity and replication topology

Overview

4.2.2 Vitals and monitoring

Alarm Monitor

View alerts for system updates, asset state changes, and errors

Alarm Monitor **Reporting**

Run and manage targeted data reports for vitals and flow activity

Reporting

4.2.3 Admin

System settings

Manage system preferences and options

System settings **Host definitions**

Create static host and subnet labels for reporting

Host definitions **User management**

Manage users, groups, and authentication

User management **Alarm configuration**

Manage alarm policies and notifications

Alarm configuration **Report management**

Manage report folders and email reports

Report management **Licensing**

Manage/view licensing details

Licensing **Instance management**

Register and manage Replicator instances

Instance management **System performance**

Monitor resource utilization and performance

System performance

4.2.3.1 Replicator UI

The Replicator web interface can be accessed from any supported browser using the server's configured IP address and the default credentials `admin:admin` and navigating to the Replicator tab/section after logging in.

> **Note**
>
> From v20.0.0 onwards, the Replicator UI/page (for both standalone deployments and local instances on Scrutinizer deployments) is accessed through the unified Plixer One/Scrutinizer web interface. The updated UI will include management functions for *"headless"* deployments and share certain components and functions (Alarm Monitor, reporting, admin menus/views, etc.) with other Plixer One platform products.

This section covers the main tabs and functionality of the Replicator UI.

On this page:

Overview *Overview* Exporters *Exporters* Profiles *Profiles* Collectors *Collectors*

Overview

The **Overview** tab of the Replicator UI can be used to view various metrics and visualizations for the system's functions.

The dashboard is updated in real time and includes the following gadgets:

- Total exporter, collector, profile, and unique exporter-collector pair counts
- Average packets in/out over time (in standard and sparkline graphs)
- Average bytes in/out over time (in standard and sparkline graphs)

Hovering over the interval markings in the sparkline graphs will display the average throughput at that point in time.

Topology

The topology diagram shows the connections between all exporters (yellow circles), profiles (green triangles), and collectors (blue squares) in a Replicator environment.

In addition, hovering over a connection in the diagram will show either of the following in the tooltip:

- Exporter to profile: The exporter as the packet/stream source and the profile (and Replicator appliance) as the destination
- Profile to collector: The profile (and Replicator appliance) as the packet/stream source and the collector as the destination

The topology diagram supports both dragging and zooming (scrolling) actions and can be forced to update at any time using the **Refresh** button.

Filters

When a filter is defined from the **Filters** menu in any of the Replicator UI views, the filter is applied to all views, including the Overview.

> **Note**
>
> The information that can be displayed in the Overview will reflect the type of filter applied:
>
> - If a profile filter is applied, only outbound/replicated traffic for the selected profile will be displayed. This is because inbound traffic vitals are not associated with a profile (i.e., inbound traffic can apply to no profiles, a single profile, or multiple profiles).
> - If a filter for a Replicator instance/appliance is applied, information for all inbound and outbound traffic on that instance will be displayed.

Exporters

Exporters are UDP-packet-exporting devices whose streams are being sent to and replicated by a Replicator server.

The **Exporters** tab of the Replicator UI shows the following details for all known exporters:

View details

- Exporter IP address
- Port packets were received on
- Current state/availability of the exporter
- Replicator server that received packets from the exporter

- Timestamp when packets were last received from the exporter
- Number of profiles whose *policies* include the exporter
- Number of collectors that have received replicated packets from the exporter
- Timestamp when details for the exporter were last modified

> **Note**
>
> The list will only include devices that have sent packets/streams to a Replicator server and are included in at least one profile's policies.

Clicking on an IP address in the main exporter list/table opens a summary tray containing further details (including matching profiles and collectors) for the exporter. Filters (Replicator server, profiles, exporters, etc.) can also be applied to the exporter list via the **Filtering Options** menu/tray.

> **Important**
>
> When using a standalone Replicator by itself, port 4739 is reserved for vitals/internal flows and cannot be used for inbound exporter streams.
>
> If there are exporters that can only send packets to Replicator over port 4739, a *headless Replicator instance* can be deployed and used to receive and replicate the streams.

Refer to the following section for further details on Replicator profiles, or see *this section* to learn more about exporter inclusion/exclusion policies.

Profiles

Replicator's packet replication functions are governed by user-defined profiles, each of which comprises the following elements:

- Profile type
- One or more exporter inclusion/exclusion policies
- One or more collectors to send replicated packets to

When a Replicator appliance receives a packet, its source is checked against all policies defined in its profiles (inclusion before exclusion). Afterwards, the packet is replicated and forwarded to all collectors assigned to matching profiles.

Profile types

A Replicator profile can enable additional roles/functions based on the selected type:

View profile types

- *IPv4 HA Dual Exporters*: Rewrites the header of IPv4 packets from a *redundant exporter pair* to show a specified IP address and port as their origin
- *IPv4 Spoofing*: Rewrites the header of IPv4 packets to show the source exporter as their origin
- *IPv6 Spoofing*: Rewrites the header of IPv6 packets to show the source exporter as their origin
- *Plixer Exporter Spoofing*: Modifies the packet header to include the origin exporter for Plixer collectors (used only in cloud environments, where conventional spoofing is not possible)

- *Auto Replicate Seed*: (Plixer One/Scrutinizer deployments only) Used to enable *automatic load-balancing* across one or more remote collectors
- *Auto Replicate Collector*: (Plixer One/Scrutinizer deployments only) Used to associate collectors with the seed profile for automatic load-balancing

> **Note**
>
> IPv4 and IPv6 spoofing profiles will ignore incoming packets in the wrong format.

High-availability exporter pairs

Replicator can automatically manage flow data streams from a specified pair of redundant IPv4 exporters using the *IPv4 Dual HA Exporters* profile type.

After a profile of this type is created, it must be configured as follows:

View instructions

1. Select the Replicator instance to use.
2. [Optional] Add a description for the profile.
3. Enter the spoofed IP address to use for replicated streams.
4. [Optional] Enter the spoofed port to use for replicated streams.
5. Create exactly two */32* policies (one for each HA exporter).
6. Select the preferred/primary source.
7. Set the amount of time to wait for the preferred source.
8. Add collectors to the profile (or define new collectors, if necessary).

After the profile has been configured and enabled, flow data from the preferred source will be replicated and forwarded to the specified collector(s). If the preferred source becomes inactive for the specified wait time, replication will start for the stream from the other exporter/policy defined in the profile. The same spoofed IP address and port will be used regardless of the active source.

Policies

Policies are exporter and port inclusion/exclusion rules that determine whether a packet should be replicated or ignored for the associated profile.

Packets/streams matching a profile's policies are replicated and forwarded to all collectors assigned to the profile. Non-matching packets are ignored but can still be replicated under other profiles.

Profile management

Replicator profiles can be created, edited, and managed from the **Profiles** tab of the Replicator UI.

Its main view comprises a table listing the following details for all existing profiles:

View details

- Profile name
- Replicator server associated with the profile
- Number of policies defined in the profile
- Number of exporters included by the profile's policies
- Number of collectors assigned to the profile
- User who created the profile
- Timestamp when the profile was last modified

Filters (Replicator server, exporters, collectors, etc.) can also be applied to the profile list via the **Filtering Options** menu/tray.

Creating a new profile

To create a new profile, click the **+** button in the main view and configure the following:

- Name for the profile
- Type of profile
- Replicator server to associate the profile with
- State (enabled or disabled)
- Description (optional)

After a new profile is saved, it is added to the main profile list/table and should be further configured to define policies and assign collectors.

> **Note**
>
> To delete one or more profiles, use the checkboxes to select them in the main view, and then use the *Delete* option in the **Bulk Actions** menu.

Editing profiles

Clicking on a profile name in the main view opens a configuration tray where policies and collectors can be added:

Adding new policies

To add a new policy:

1. Expand the **Policies** section, and then click the **+** button.
2. In the secondary tray, configure the CIDR, port, and type (*Include* or *Exclude*) for the policy.
3. Click **Save**.

Existing policies can also be edited or deleted from the policy list.

Adding new collectors

1. Expand the **Collectors** section, and then click the edit/pencil button.
2. In the secondary tray, use the checkboxes to select the collectors to assign to the policy.

Additional collectors can be *defined* by clicking the **+** button instead. To delete currently assigned collectors, click the corresponding delete icon in the list.

Basic profile settings (type, description, etc.) can also be edited in the same tray.

> **Note**
>
> A profile can include as many exporter ("in") ports and collector ("out") ports as needed, but the same port cannot be defined for both receiving and transmitting.

Collectors

Collectors are recipient hosts that are assigned to profiles to configure the destinations (by IP address and port) for replicated packets. When a packet matching a profile's *exporter policies* is received, it is replicated and forwarded to all collectors assigned to the profile.

A collector must be defined before it can be assigned to a profile. Once saved, collectors can be assigned to any number of profiles as needed.

Collector management

Replicator collector definitions can be added, modified, and managed from the **Collectors** tab of the Replicator UI.

Its main view comprises a table listing the following details for all defined collectors:

View details

- Collector IP address
- Port used to send replicated packets
- Current state/availability of the collector
- Replicator server associated with the collector
- Timestamp when the collector was last confirmed as available
- Number of profiles the collector is currently assigned to
- Number of exporters the collector has received packets from
- Timestamp when the collector definition was last modified

Filters (Replicator server, profiles, exporters, etc.) can also be applied to the collector list via the **Filtering Options** menu/tray.

Adding a new collector

To add a new collector definition, click the **+** button in the main view and configure the following:

- Collector IP address
- Port to send replicated packets on
- Replicator server to associate with the collector
- Description (optional)

After a new collector definition is saved, it is added to the main collector list/table. Clicking on a collector IP address opens a configuration tray where it can be assigned to profiles. Settings for the collector can also be modified from this tray at any time.

> **Note**
>
> To delete one or more collector definitions, use the checkboxes to select them in the main view, and then use the *Delete* option in the **Bulk Actions** menu.

Assigning collectors to profiles

Collectors can be assigned to profiles from either the collector configuration tray or the *profile configuration tray*.

In the collector configuration tray, click the edit/pencil button in the **Profiles** section, and then select all profiles to assign the collector to in the secondary tray.

> **Note**
>
> The profile assignment tray for a collector can be accessed directly from the three-dot/overflow menu in the main view.

4.2.3.2 System management

> **Note**
>
> For detailed guides on these functions, see this section of the Plixer One/Scrutinizer manual.

On this page:

Alarm Monitor

The **Alarm Monitor** view alerts users to certain Replicator actions, behaviors, and state changes using predefined alarm policies.

After an event has been addressed, it can be hidden/dismissed by selecting it and clicking **Acknowledge Selected Events**.

> **Note**
>
> To get alerted to alarms/events through other channels, *create notification profiles* and assign them to alarm policies as needed.

Events by policy

The **Alarm Monitor > Policies** subview will show all alarm policies with active/unacknowledged events. Drilling into a policy opens a summary page where individual event artifacts can be inspected for further details.

Events by host

The **Alarm Monitor > Hosts** subview will list all hosts associated with active/unacknowledged events. Drilling into a host opens a summary page where alarm policies triggered by that host can be inspected for further details.

Alarm policies

The following alarm policies are used to monitor activity for Replicator events:

View list

- *Auto Replicate Error* - Reports unexpected irregularities related to autoreplication
- *Auto Replicate Exporter Added* - Reports exporters being added to an Auto Replicate profile
- *Auto Replicate Exporter Removed* - Reports exporters being removed from an Auto Replicate profile
- *Auto Replicate Ran* - Reports Auto Replicate being run
- *HA Exporter switchover event* - Reports active exporter changes in Replicator profiles
- *Replicator Collector State Change* - Reports changes in collector states
- *Replicator Exporter State Change* - Reports changes in exporter states
- *Replicator Has Encountered an Error* - Reports general Replicator errors
- *Replicator High Availability State Change* - Reports high availability state changes for Replicator instances

Note

- Exporter and collector states are also displayed in the main list/table of their respective tabs.
- The Alarm Monitor views will also report events associated with system vitals and general activity.

Reporting

The **Reports** page can be used to run and manage targeted data reports for Replicator vitals and flow activity.

Creating/running a report

To create/run a new report, navigate to the **Reports > Run Report** page and follow these steps:

View instructions

1. Select between the two options to start configuring a report:
   - **Select Devices**: Select one or more devices to use as data sources for the report before specifying the report type.
   - **Select Report Type**: Select a report type to define the data aggregation criteria before specifying data sources.
2. After the devices and report type have been selected, configure the following settings/filters for the report:
   - **Time Window**: Select a *Last* **X** time window or specify a custom range to be covered by the report (default: *last 24 hours*).
   - **Display Type**: Select the graph or chart for result visualization in the output view.
   - **Additional Filters**: Define any additional filters to be applied to the report.

3. Click **Run Report**.

After the report completes running, the report configuration can be saved and/or further modified directly from the output view.

Saved reports

After a report has been created, it can be saved and re-run at any time from the **Reports > Saved Reports** subtab. Saved reports can also be assigned to report folders in this view.

To create or manage report folders, click the folder icon or any entry in the *Folders* column of the main view.

Scheduled email reports

The **Reports > Scheduled** subtab can be used to set up scheduled email reports, which run a specified saved report at regular intervals and email the results to one or more email addresses. Scheduled email reports can also be configured to include multiple reports as well as PDF and/or CSV copies of the output.

To create a new scheduled email report, click on the **+** button and configure the desired settings in the tray.

> **Note**
>
> Saved report folder and scheduled report management functions can also be accessed via the *Admin > Reports* section.

Admin

The **Admin** section of the web interface provides access to administrative and configuration functions for Replicator appliances.

> **Note**
>
> The admin menus/views for local Replicator instances on Scrutinizer deployments are part of a unified **Admin** section for Scrutinizer and other Plixer One Platform products.

System settings

The **Admin > Settings** menu can be used to manage system preferences and options for Replicator.

View list

- **DNS** - Set DNS cache retention duration and resolution attempt timeout
- **Data History** - Set alarm and flow data history retention durations
- **Global Authentication Settings** - Configure user session and login security options
- **Login Banner** - Add a custom message to the Replicator login page
- **Reporting** - Customize Replicator reporting engine functions
- **System Preferences** - Configure general Replicator environment preferences/settings
- **System/New User Default** - Set up default preferences/settings for new users

Host definitions

The **Admin > Definitions** menu can be used to create static host and subnet labels for reporting.

User management

The **Admin > Users & Groups** menu can be used to manage user accounts, user groups, and authentication options.

- **Auditing Logs** - View logs of web interface user actions
- **Authentication Providers** - Add and configure third-party authentication methods/servers
- **Authentication Settings** - Configure global options for local and third-party authentication methods
- **Authentication Tokens** - Add and manage user authentication tokens
- **User Accounts** - Manage user accounts and preferences
- **User Groups** - Set up local user groups and manage access to features and resources

Alarm configuration

The **Admin > Alarm Monitor** menu can be used to manage alarm policies as well as create and assign notification profiles.

Alarm policies

The **Alarm Policies** admin view lists all policies used to monitor system activity for alarms/events.

Clicking on an alarm policy opens a tray containing event/message details and configuration options (enable/disable, timeout for discrete events, etc.) for that policy. One or more custom notification profiles can also be assigned to the policy from this tray.

Notification profiles

The **Notification Profiles** admin view allows users to create and manage notification profiles, which can be used to add custom notifications actions to alarm policies.

To learn more about creating notification profiles and the different notification types, see this topic in the Plixer One/Scrutinizer manual.

Report management

The **Admin > Reports** menu contains the following options:

- **Report Folders** - Create and manage folders for *saved reports*
- **Scheduled Email Reports** - Set up and manage *scheduled email report* configurations

Licensing

The **Admin > Plixer** menu can be used to add a new Replicator license key or view details for the active license.

> **Note**
>
> To obtain a new license key, contact *Plixer Technical Support* and provide the *Machine ID* displayed in the **Admin > Plixer > Replicator** view.

A standalone Replicator server can also be upgraded to a Plixer One/Scrutinizer deployment by adding an active license key in the **Admin > Plixer > Scrutinizer** view.

Instance management

The **Admin > Resources > Replicators** page is used to configure and manage individual Replicator instances/appliances. Additional headless appliances must also be registered from this page before being *deployed*.

The main view of this page lists the following details for the main/standalone Replicator instance (or the local instance on a Scrutinizer deployment) and any additional headless instances registered:

- Name assigned to the instance
- Hostname or IP address
- License status
- Authentication token (required to deploy the appliance)
- Deployment type (*Primary* or *Secondary* for *instances paired for high availability*, *Single* for unpaired deployments)
- Paired primary or secondary instance (high availability only)
- User who deployed the instance
- Timestamp when the instance was registered

Clicking on an instance name opens a configuration tray where the settings (see below) for that instance can be configured.

> **Note**
>
> If the main/local instance is deleted, it can be re-added by clicking on the **Add** (**+**) button in the main view and then selecting *Add Local Replicator*.

Replicator settings

The following settings can be modified via the configuration tray for each Replicator instance:

- **Name**: Name assigned to the instance/appliance
- **Ping Collectors**: Enable to periodically ping collectors to confirm availability
- **Stop Replicator**: Enable to automaticaly stop packet replication/forwarding to collectors that are down (unavailable if *Ping Collectors* is disabled)
- **Stop Replication Timeout**: Number of minutes a collector must be down before replication is stopped
- **High Availability**: Enable to pair a secondary Replicator instance for high availability (see *this guide* for further details)

The configuration tray also includes a **View Replicator** shortcut, which applies a filter for the selected instance to the main Replicator UI/page.

Registering a new instance

To register/add a new headless Replicator instance, follow these steps:

> **Note**
>
> A headless instance must be registered before the appliance is deployed.

View instructions

1. Click the **+** button in the main view.
2. Enter a name for the headless instance to be deployed.
3. Click the **Save** button.

After an instance has been registered, note the authentication in the configuration tray, and then *deploy the headless appliance/VM*.

System performance

The **Admin > Resources > System Performance** page can be used to monitor resource utilization and performance for the standalone Replicator appliance.

Drilling down into a collector from the summary table opens a more detailed view with current and predicted disk utilization based on total disk capacity and current flow volume.

4.3 Advanced Services

High availability

Set up redundant Replicator instances for fault tolerance

High availability **Auto Replicate**

Automatic stream management for multiple collectors

Auto Replicate **Replicator APIs**

Leverage Replicator APIs for external integration

Replicator APIs **Version upgrades**

Upgrade procedures and instructions

Version upgrades **Profile migration**

Migrate profile data to v20.0.0+ deployments

Profile migration

4.3.1 High availability

A Replicator instance can be paired with a secondary instance to create a high-availability pair that ensures uninterrupted flow data replication. Configuration data is synced between the primary and secondary instances for seamless failover.

> **Note**
>
> Any unpaired *headless deployment* without saved profiles can be set as a secondary instance in an HA pair.

4.3.1.1 Multi-network configuration

When the primary and secondary Replicator instances are on different subnets (i.e., a virtual IP address cannot be used), the flow data to be replicated must be sent to both Replicator instances.

After a high availability pair is set up in this mode, the primary instance continuously sends UDP heartbeat packets to the secondary instance (1 packet per second). If the secondary fails to receive two consecutive heartbeat packets, it immediately starts replication. Once a heartbeat packet is received from the primary instance again, the secondary syncs any configuration updates, stops replication, and reverts to the standby state.

> **Note**
>
> Multi-network mode is the default high-availability configuration. If the primary and secondary Replicator instances are on the same network, enabling virtual IP/single-network mode is recommended.

Enabling multi-network HA on a Replicator

To create a multi-network HA pair, follow these steps:

View instructions

1. Go to *Admin* > *Resources* > *Replicators*, and then click on the name of the Replicator instance to use as the primary.
2. In the Replicator configuration tray, toggle the **High Availability** switch to *On* (will not be displayed if no secondary instances are available).
3. Select the Replicator instance to use as the secondary in the *Secondary IP* dropdown.
4. Click **Save** to create the HA pair.

After the multi-network HA pair has been saved, configure all exporters to send flow data to both the primary and secondary instances.

4.3.1.2 Single-network configuration

When the primary and secondary Replicator instances are on the same network, they can receive flow data packets via a shared virtual IP address.

After a high-availability pair is set up in this mode, the availability of the primary and secondary instances is monitored using the Virtual Router Redundancy Protocol (VRRP). If the primary Replicator instance becomes unavailable, the specified virtual IP address is immediately reassigned to the secondary instance, which then starts replication (handover typically takes ~1 second). Once the primary becomes available again, it re-assumes responsibility for the IP address and resumes replication after a user-defined delay (see instructions below).

Enabling virtual IP HA on a Replicator

View instructions

1. Go to *Admin* > *Resources* > *Replicators*, and then click on the name of the Replicator instance to use as the primary.
2. In the Replicator configuration tray, toggle the **High Availability** switch to *On*.

3. Select the Replicator instance to use as the secondary in the *Secondary IP* dropdown.
4. Enable *Virtual IP*, and then enter the following details in the provided fields:
    - Virtual IP address: IP address to be shared between the primary and secondary instances
    - Virtual router ID: Virtual router ID to assign to the HA pair (must be unique to the pair to avoid conflicts with other software devices using VRRP)
    - Failover delay: Length of time that the primary instance must be online again before it takes over the virtual IP and replication (to avoid flapping)

    Note

    The failover delay is meant to allow all services on primary instance to fully restart after a reboot/outage. A delay of at least 2 minutes is recommended (default: 5 minutes).

5. Click **Save** to create the HA pair.

After the VIP HA pair has been saved, configure all exporters to send flow data to the virtual IP address specified.

4.3.1.3 Reverting HA pairings

To revert paired Replicator instances back to single appliances, toggle off *High Availability* for the primary instance in the *Admin > Resources > Replicators* configuration tray.

This will unpair the instances and allow them to be used as single deployments again. The primary instance will retain all profiles, collectors, and other settings previously applied, and the secondary instance will be reverted to its unused, post-deployment state.

4.3.2 Auto Replicate

Replicator **Auto Replicate** allows flow streams to all be sent to a single Replicator instance, which will then automatically distribute them across collectors in the cluster based on their available capacity.

Auto Replicate is enabled by creating a collector profile for each destination collector and associating them with a seed profile with the necessary exporter inclusion/exclusion policies. The profiles must be created on the local Replicator instance on a Scrutinizer deployment or a *headless instance* registered with the primary reporter. Destination collectors for autoreplication must be part of the same cluster, whose primary reporter will have access to all collector configurations and current loads.

Note

Multiple seed profiles can be created to enable autoreplication for separate collector groups. All seed profiles are automatically discovered and processed when rebalancing and assigning exporters.

4.3.2.1 Creating collector profiles

Auto Replicate Collector profiles define the destination collectors for a seed profile. A collector profile must be created for each collector and then assigned to the seed profile for its cluster.

To create a new collector profile:

View instructions

1. Navigate to **Replicator > Collectors**, and then click the **+** icon to create a new collector profile.
2. In the *Add Replicator Profile* tray, enter a name for the profile.
3. Select **Auto Replicate Collector** as the profile type, and then select the Replicator instance to create the profile on.
4. Enter the collector's IP address and port number to use.
5. Enter an exporter count limit and a flow rate limit for the collector.
6. [OPTIONAL] Add a description for the collector/profile.
7. Click **Save**.

Repeat the above steps to create a collector profile for each destination collector, and then proceed to create the seed profile.

4.3.2.2 Creating the seed profile

The Auto Replicate Seed profile contains *inclusion and exclusion policies* that define the exporters/streams that should be autoreplicated.

Flows from matching exporters are sent to one of the collectors defined by the collector profiles associated with the seed profile. Each new exporter is always assigned to the collector with the most available capacity. If a collector becomes overloaded (based on exporter count or flow rate) as a result, the exporter will be reassigned to a collector with the required capacity available.

To create the seed profile:

View instructions

1. Navigate to **Replicator > Profiles**, and then click the **+** icon to *create a new profile*.
2. In the *Add Replicator Profile* tray, enter a name for the profile.
3. Select **Auto Replicate Seed** as the profile type in the dropdown, and then select the Replicator instance to create the profile on.
4. [OPTIONAL] Add a description for the collector/profile.
5. Click **Save**, and then return to the main **Profiles** view.
6. Click on the newly created profile to open the configuration tray.
7. Create inclusion and exclusion policies to define source exporters for autoreplication.
8. Select the collector profiles of all destination collectors to associate with the seed profile (only collector profiles not currently associated with a seed profile can be selected).
9. Click **Save**.

Once the seed profile has been configured, enabling it will start autoreplication.

> **Note**
>
> By default, new exporters/streams are assigned to collectors once every hour, and collectors are checked to verify that they are not over capacity once a day. These times can be adjusted in `/home/plixer/scrutinizer/files/conf/rebalance.yaml`. Rebalancing can also be manually initiated via the seed profile in the Replicator UI. Exporter reassignment is kept to a minimum to improve system performance.

Managing collector assignments

The *Collector Profiles* section of the configuration tray can be used to add or remove collector profiles from the seed profile.

When adding collector profiles, only profiles that are currently unlinked (created without a seed profile association or removed from another seed profile) will be available.

4.3.3 Replicator APIs

Changed in version 20.0.0: The Replicator API has been completely redesigned in version 20.0.0 to align with the Plixer One Platform architecture. The previous REST-style `/api/1/*` endpoints are no longer available. All API calls now use RESTful paths under `/api/v2/replicator/`.

On this page:

4.3.3.1 API Overview

Base URL

```
https://[hostname]/api/v2/replicator/
```

The REST API uses standard HTTP methods:

Method	Action
`GET`	List/search resources
`POST`	Create resources
`PUT`	Update resources
`DELETE`	Delete resources

Endpoints:

Endpoint	Description
`/api/v2/login`	POST to authenticate
`/api/v2/logout`	POST to end session
`/api/v2/replicator/deployments`	List Replicator deployments
`/api/v2/replicator/profiles`	List/create profiles
`/api/v2/replicator/profiles/:id`	Get/update/delete profile
`/api/v2/replicator/profiles/:id/collectors`	Collectors for profile
`/api/v2/replicator/profiles/:id/exporters`	Exporters for profile (read-only)
`/api/v2/replicator/collectors`	List/create collectors
`/api/v2/replicator/exporters`	List exporters (read-only)
`/api/v2/replicator/policies/:profile_id`	Policies for profile

Response Format

All responses are JSON objects. Successful list responses include pagination metadata:

```
{
  "results": [...],
  "totalRowCount": 100,
  "rowCount": 10,
  "offset": "0",
  "allLoaded": 0
}
```

Field	Description
`results`	Array of matching data rows
`totalRowCount`	Total count of all matching rows across all pages
`rowCount`	Number of rows in this response
`offset`	Starting position of this response
`allLoaded`	1 when all rows have been returned, `0` otherwise

Error responses include:

```
{
  "err": "error_code",
  "details": "Additional error information"
}
```

Pagination

Uses query parameters to paginate list endpoints.

Parameter	Type	Default	Description
`maxRows`	integer	400	Maximum number of rows to return per request
`offset`	integer	0	Starting row position (0-based)
`page`	integer	—	Page number (1-based). Alternative to `offset`; when used, offset is calculated as `(page - 1) * maxRows`

Example — first page of 10 results:

```
curl -X GET 'https://[hostname]/api/v2/replicator/exporters?replicator_ip=10.42.100.142&
↪maxRows=10&offset=0' \
  -b cookies.txt \
  -k
```

Example — second page of 10 results:

```
curl -X GET 'https://[hostname]/api/v2/replicator/exporters?replicator_ip=10.42.100.142&
↪maxRows=10&offset=10' \
  -b cookies.txt \
  -k
```

Example — using page number instead of offset:

```
curl -X GET 'https://[hostname]/api/v2/replicator/exporters?replicator_ip=10.42.100.142&
↪maxRows=10&page=2' \
  -b cookies.txt \
  -k
```

To iterate through all results, increment `offset` by `maxRows` until `allLoaded` is 1 or `offset` >= `totalRowCount`.

Key Differences from v19

Aspect	v19 (Legacy)	v20 (Current)
Endpoint	`GET/POST /api/1/[resource]/[action]/...`	REST: `/api/v2/replicator/[resource]`
Profile ports	Profiles had `listeningport` and `sendingport`	Ports are now on collectors and exporters
Exporter assignment	Direct assignment to profiles	Exporters are linked via policies (CIDR-based matching)
Authentication	`/api/1/login/` with SHA3 password	Session-based authentication with plaintext password over HTTPS
Multi-Replicator	Single appliance per API	Manage multiple Replicators from one interface via `replicator_ip`

4.3.3.2 Authentication

Replicator uses session-based authentication. You must authenticate to obtain a session ID, then include the session cookies in subsequent requests.

Login

Authenticates a user and establishes a session.

Login API call

```
curl -X POST 'https://[hostname]/api/v2/login' \
  -d 'name=admin' \
  -d 'pwd=yourpassword' \
  -c cookies.txt \
  -k
```

Parameter	Description
`name`	Username
`pwd`	Password (plaintext - always use HTTPS)

Successful response:

```
{
  "sessionid": "MQJK0iTn2ckxv5tK",
  "userid": "1",
```

(continues on next page)

(continued from previous page)

```
  "csrfToken": "jEHJdqWbRAR7B4OM"
}
```

Error response:

```
{
  "err": "loginFailed"
}
```

Important

Store the session cookies returned by the server using `-c cookies.txt`. Include them in all subsequent API requests using `-b cookies.txt`.

Logout

Terminates the current session.

Logout API call

```
curl -X POST 'https://[hostname]/api/v2/logout' \
  -b cookies.txt \
  -k
```

4.3.3.3 Replicator Deployments

Version 20.0.0 introduces the ability to manage multiple Replicator appliances from a single interface. Each Replicator deployment is identified by its IP address (`replicator_ip`), which must be specified when managing profiles, collectors, and policies.

List Replicator Deployments

Retrieves all configured Replicator deployments.

List deployments API call

```
curl -X GET 'https://[hostname]/api/v2/replicator/deployments' \
  -b cookies.txt \
  -k
```

Response:

```
{
  "results": [
    {
      "deployment_id": 1,
      "replicator_name": "Local",
      "replicator_ip": "10.42.100.142",
      "replicator_dns": "10.42.100.142",
```

(continues on next page)

(continued from previous page)


```
      "deployment_type": "single",
      "is_primary": 1,
      "license": 1,
      "auto_replicate": 0,
      "ping_collectors": 1,
      "stop_replicator": 0,
      "created_at_label": "2025-08-12 19:07",
      "created_by": 1,
      "uname": "admin",
      "paired_ip": null,
      "paired_dns": null,
      "paired_replicator_name": null,
      "virtual_ip": null,
      "virtual_ip_dns": null,
      "virtual_router_id": null,
      "failover_delay": null
    }
  ],
  "totalRowCount": 1
}
```


Deployment Types:

- `single` - Standalone Replicator
- `primary` - Primary in an HA pair
- `secondary` - Secondary in an HA pair

Using replicator_ip

The `replicator_ip` parameter identifies which Replicator deployment to operate on.

Required for all write operations (POST, PUT, DELETE) on profiles, collectors, and policies. Omitting it from create or update requests will result in the resource being created without a Replicator association.

Recommended for all read operations (GET) to scope results to a specific Replicator deployment.

```
# List profiles for a specific Replicator
curl -X GET 'https://[hostname]/api/v2/replicator/profiles?replicator_ip=10.42.100.142' \
  -b cookies.txt \
  -k
```

4.3.3.4 Profile Management

Profiles are used to define replication configurations.

> **Note**
>
> In v20, port configuration has been removed from profiles and are now defined at the collector and exporter level instead.

Profile Types

Type	Description
ipv4_spoofing	Standard IPv4 replication profile
ipv6_spoofing	Standard IPv6 replication profile
plixer_host_tag	Host tag-based replication
auto_replicate_seed	Seed profile for Auto-Replicate feature
auto_replicate_collector	Collector profile for Auto-Replicate
ha_dual_exporters_profile	High availability dual exporter profile

List Profiles

Retrieves a list of profiles with optional filtering and pagination.

List profiles API call

```
curl -X GET 'https://[hostname]/api/v2/replicator/profiles?replicator_ip=10.42.100.142' \
  -b cookies.txt \
  -k
```

Optional Query Parameters:

Parameter	Description
replicator_ip	Filter by Replicator IP address
name	Filter by profile name (supports partial match)
profile_type	Filter by profile type

Response:


```
{
  "results": [
    {
      "profile_id": 19,
      "profile_name": "Tester",
      "profile_type": "ipv4_spoofing",
      "profile_type_langed": "IPv4 Spoofing",
      "description": "",
      "enabled": 1,
      "replicator_ip": "10.42.100.142",
      "replicator_name": "Tester",
      "replicator_dns": "10.42.100.142",
      "collectors": 0,
      "exporters": 0,
      "policies": 0,
      "igroup_policies": 0,
      "profile_options": null,
      "created_by": 1,
      "uname": "admin",
      "modified_by": null,
      "modified_by_name": null,
```


(continues on next page)

(continued from previous page)


```
      "modified_ts": "1765228249.238230",
      "modified_date": "2025-12-08 21:10",
      "seed_profile": null,
      "virtual_ip": null,
      "virtual_ip_dns": null
    }
  ],
  "totalRowCount": 1,
  "rowCount": 1
}
```


Create Profile

Creates a new replication profile.

Create profile API call

```
curl -X POST 'https://[hostname]/api/v2/replicator/profiles' \
  -d 'replicator_ip=10.42.100.142' \
  -d 'profile_name=My_New_Profile' \
  -d 'profile_type=ipv4_spoofing' \
  -d 'description=Created via API' \
  -d 'enabled=1' \
  -b cookies.txt \
  -k
```

Parameters:

Parameter	Description
`replicator_ip`	IP address of the Replicator deployment (required)
`profile_name`	Name for the new profile (required)
`profile_type`	Profile type (default: `ipv4_spoofing`)
`description`	(Optional) Profile description
`enabled`	(Optional) `1` to enable, `0` to disable (default: `1`)

Successful response:


```
{
  "id": 21
}
```


Update Profile

Updates an existing profile.

Update profile API call

```
curl -X PUT 'https://[hostname]/api/v2/replicator/profiles/21' \
  -d 'replicator_ip=10.42.100.142' \
  -d 'enabled=0' \
```

(continues on next page)

(continued from previous page)

```
  -b cookies.txt \
  -k
```

Parameters:

Parameter	Description
`profile_id`	ID of the profile to update (in URL path)
`replicator_ip`	IP address of the Replicator deployment (required)
`profile_name`	(Optional) New profile name
`description`	(Optional) New description
`enabled`	(Optional) `1` to enable, `0` to disable

Successful response:

```
{
  "id": "21"
}
```

Delete Profile

Deletes one or more profiles.

Delete profile API call

```
curl -X DELETE 'https://[hostname]/api/v2/replicator/profiles/21' \
  -d 'json=[21]' \
  -b cookies.txt \
  -k
```

Parameters:

Parameter	Description
`json`	JSON array of profile IDs to delete

Successful response:

```
{
  "success": 1
}
```

4.3.3.5 Collector Management

Collectors are destination hosts that receive replicated packet streams. Each collector is identified by an IP address and port combination.

List Collectors

Retrieves a list of collectors with their status.

List collectors API call

```
curl -X GET 'https://[hostname]/api/v2/replicator/collectors?replicator_ip=10.42.100.142
↪' \
  -b cookies.txt \
  -k
```

Optional Query Parameters:

Parameter	Description
`replicator_ip`	Filter by Replicator IP
`collector_ip`	Filter by collector IP
`collector_port`	Filter by collector port

Response:


```
{
  "results": [
    {
      "collector_ip": "10.42.100.142",
      "collector_port": 4739,
      "collector_dns": "10.42.100.142",
      "replicator_ip": "10.42.100.142",
      "replicator_dns": "10.42.100.142",
      "note": "",
      "mtu": 1500,
      "profiles": 1,
      "exporters": 15,
      "state": "active",
      "details": "collector appears to be active",
      "lastup": 1768859020,
      "last_seen": "2026-01-19 21:43",
      "modified_ts": "1768859041.802129",
      "modified_date": "2026-01-19 21:44"
    }
  ],
  "totalRowCount": 1
}
```


Collector States:

- `active` - Collector is receiving traffic
- `null` - Collector has not been seen yet

Create Collector

Creates a new collector.

Create collector API call

```
curl -X POST 'https://[hostname]/api/v2/replicator/collectors' \
  -d 'replicator_ip=10.42.100.142' \
  -d 'collector_ip=10.1.10.60' \
  -d 'collector_port=2055' \
  -d 'note=Primary collector' \
  -b cookies.txt \
  -k
```

Parameters:

Parameter	Description
`replicator_ip`	Replicator IP address (required)
`collector_ip`	Collector IP address (required)
`collector_port`	Collector UDP port (required)
`note`	(Optional) Description or note

Assign Collector to Profile

Associates a collector with a profile.

Assign collector to profile API call

```
curl -X POST 'https://[hostname]/api/v2/replicator/profiles/19/collectors' \
  -d 'replicator_ip=10.42.100.142' \
  -d 'collectors=[{"collector_ip":"10.1.10.60","collector_port":2055}]' \
  -b cookies.txt \
  -k
```

Parameters:

Parameter	Description
`profile_id`	Profile ID (in URL path)
`replicator_ip`	IP address of the Replicator deployment (required)
`collectors`	JSON array of collector objects (required)

List Profile Collectors

Lists collectors with their assignment status for a profile.

List profile collectors API call

```
curl -X GET 'https://[hostname]/api/v2/replicator/profiles/5/collectors' \
  -b cookies.txt \
  -k
```

Response:


```
{
  "results": [
    {
      "collector_ip": "10.42.100.142",
      "collector_port": 4739,
      "collector_dns": "10.42.100.142",
      "replicator_ip": "10.42.100.142",
      "added_to_profile": 1,
      "state": "active",
      "details": "collector appears to be active",
      "lastup": 1768859020,
      "modified_date": "2026-01-19 21:44"
    },
    {
      "collector_ip": "10.42.100.143",
      "collector_port": 4739,
      "collector_dns": "10.42.100.143",
      "replicator_ip": "10.42.100.142",
      "added_to_profile": 0,
      "state": "active",
      "details": "collector appears to be active"
    }
  ],
  "totalRowCount": 2
}
```


The `added_to_profile` field indicates whether the collector is assigned to the specified profile (`1`) or not (`0`).

Delete Collector

Removes a collector.

Delete collector API call

```
curl -X DELETE 'https://[hostname]/api/v2/replicator/collectors' \
  -d 'json=[{"collector_ip":"10.1.10.60","collector_port":2055,"replicator_ip":"10.42.
↪100.142"}]' \
  -b cookies.txt \
  -k
```

Parameters:

Parameter	Description
`json`	JSON array of collector objects to delete

4.3.3.6 Exporter Status

Exporters are network devices that send flow data to the Replicator. Unlike collectors, exporters are not created manually. Exporters are automatically discovered when they begin sending traffic to the appliance. The following Exporter APIs can be used to view exporter status and the profiles they are associated with via policies.

List Exporters

Retrieves all discovered exporters and their current status.

List exporters API call

```
curl -X GET 'https://[hostname]/api/v2/replicator/exporters?replicator_ip=10.42.100.142'␣
↪\
  -b cookies.txt \
  -k
```

Optional Query Parameters:

Parameter	Description
`replicator_ip`	Filter by Replicator IP

Response:


```
{
  "results": [
    {
      "exporter_ip": "10.1.1.70",
      "exporter_port": 2056,
      "exporter_dns": "DevSuiteC3650.plxr.local",
      "replicator_ip": "10.42.100.142",
      "replicator_dns": "10.42.100.142",
      "note": null,
      "profiles": 3,
      "collectors": 2,
      "state": "active",
      "details": "replicator has received UDP traffic from exporter within the last 5␣
↪minutes",
      "lastheard": 1768858941,
      "last_seen": "2026-01-19 21:42",
      "modified_ts": "1768859010.178782",
      "modified_date": "2026-01-19 21:43"
    },
    {
      "exporter_ip": "10.100.77.2",
      "exporter_port": 2055,
      "exporter_dns": "10.100.77.2",
      "replicator_ip": "10.42.100.142",
      "replicator_dns": "10.42.100.142",
      "note": null,
      "profiles": 2,
      "collectors": 1,
      "state": "unknown",
```


(continues on next page)

(continued from previous page)


```
      "details": "replicator has not received UDP traffic from exporter in over 5 minutes
↪",
      "lastheard": 1768856045,
      "last_seen": "2026-01-19 20:54"
    }
  ],
  "totalRowCount": 36
}
```


Exporter States:

- `active` - Replicator has received traffic from this exporter within the last 5 minutes
- `unknown` - Replicator has not received traffic from this exporter in over 5 minutes

Key Fields:

- `profiles` - Number of profiles this exporter is associated with (via policies)
- `collectors` - Number of collectors receiving replicated traffic from this exporter
- `lastheard` - UNIX timestamp of when traffic was last received
- `last_seen` - Human-readable timestamp of when traffic was last received

Pagination example — page through exporters 10 at a time:

```
# First page
curl -X GET 'https://[hostname]/api/v2/replicator/exporters?replicator_ip=10.42.100.142&
↪maxRows=10&offset=0' \
  -b cookies.txt -k

# Second page
curl -X GET 'https://[hostname]/api/v2/replicator/exporters?replicator_ip=10.42.100.142&
↪maxRows=10&offset=10' \
  -b cookies.txt -k
```

List Profile Exporters

Retrieves exporters that are associated with a specific profile (via policies).

List profile exporters API call

```
curl -X GET 'https://[hostname]/api/v2/replicator/profiles/5/exporters' \
  -b cookies.txt \
  -k
```

Note: The profile ID is required in the URL path.

Response:


```
{
  "results": [
    {
      "exporter_ip": "10.1.1.70",
      "exporter_port": 2056,
```


(continues on next page)

(continued from previous page)


```
      "exporter_dns": "DevSuiteC3650.plxr.local",
      "replicator_ip": "10.42.100.142",
      "state": "active",
      "details": "replicator has received UDP traffic from exporter within the last 5␣
↪minutes",
      "lastheard": 1768858941,
      "last_seen": "2026-01-19 21:42"
    }
  ],
  "totalRowCount": 15
}
```


This endpoint returns only exporters that match the policies configured for the specified profile, making it useful for understanding which flow sources are being replicated by a particular profile.

4.3.3.7 Policy Management

Policies define which exporters are associated with a profile.

> **Note**
>
> In v20, exporters can no longer be directly assigned to profiles. They are linked through CIDR-based policies that match exporter IP addresses.

Policy Model

Field	Description
`profile_id`	The profile that the policy belongs to
`ip_range`	CIDR notation for matching exporter IPs (e.g., `10.1.1.0/24` or `10.1.1.1/32` for a single IP)
`received_port`	The UDP port this policy applies to
`include`	`1` for include policy, `0` for exclude policy

List Policies

Retrieves policies for a profile.

List policies API call

```
curl -X GET 'https://[hostname]/api/v2/replicator/policies/1' \
  -b cookies.txt \
  -k
```

> **Important**
>
> The profile ID is required in the URL path.

Response:

```
{
  "results": [
    {
      "profile_id": 1,
      "ip_range": "0.0.0.0/0",
      "received_port": 2055,
      "include": 1
    },
    {
      "profile_id": 1,
      "ip_range": "0.0.0.0/0",
      "received_port": 2056,
      "include": 1
    }
  ],
  "totalRowCount": 2
}
```


Create Policy

Creates a new policy to associate exporters with a profile.

Create policy API call

Include all exporters on port 2055:

```
curl -X POST 'https://[hostname]/api/v2/replicator/policies/19' \
  -d 'ip_range=0.0.0.0/0' \
  -d 'received_port=2055' \
  -d 'include=1' \
  -b cookies.txt \
  -k
```

Parameters:

Parameter	Description
`profile_id`	Profile ID (in URL path)
`ip_range`	CIDR notation (required, e.g., `10.1.1.0/24`)
`received_port`	UDP port (required)
`include`	1 for include, `0` for exclude (required)

Example - include a specific subnet:

```
curl -X POST 'https://[hostname]/api/v2/replicator/policies/19' \
  -d 'ip_range=192.168.1.0/24' \
  -d 'received_port=2055' \
  -d 'include=1' \
  -b cookies.txt \
  -k
```

Example - include a single exporter IP:

```
curl -X POST 'https://[hostname]/api/v2/replicator/policies/19' \
  -d 'ip_range=10.1.1.1/32' \
  -d 'received_port=2055' \
  -d 'include=1' \
  -b cookies.txt \
  -k
```

Successful response:

```
{
  "success": 1,
  "policyCount": 1
}
```

Delete Policy

Removes policies from a profile.

Delete policy API call

```
curl -X DELETE 'https://[hostname]/api/v2/replicator/policies/19' \
  -d 'json=[{"ip_range":"192.168.1.0/24","received_port":2055}]' \
  -b cookies.txt \
  -k
```

Parameters:

Parameter	Description
`profile_id`	Profile ID (in URL path)
`json`	JSON array of policy objects to delete

Successful response:

```
{
  "success": 1
}
```

4.3.3.8 Common Workflows

Creating a Complete Replication Setup

To set up replication from exporters to collectors, do the following:

1. **Authenticate** to obtain a session.
2. **Create a profile** with the desired type.
3. **Create a collector** with IP and port.
4. **Assign the collector to the profile**.
5. **Create policies** to match exporters by CIDR range.

Example: Complete Setup Script

```
#!/bin/bash
HOST="https://[hostname]"
AUTH="$HOST/api/v2"
API="$HOST/api/v2/replicator"
REPLICATOR_IP="10.42.100.142"

# 1. Authenticate
curl -s -X POST "$AUTH/login" \
  -d 'name=admin' \
  -d 'pwd=yourpassword' \
  -c cookies.txt -k

# 2. Create a profile
PROFILE_RESPONSE=$(curl -s -X POST "$API/profiles" \
  -d "replicator_ip=$REPLICATOR_IP" \
  -d 'profile_name=API_Created_Profile' \
  -d 'profile_type=ipv4_spoofing' \
  -d 'description=Created via API' \
  -b cookies.txt -k)

PROFILE_ID=$(echo $PROFILE_RESPONSE | python3 -c "import sys,json; print(json.load(sys.
↪stdin)['id'])")
echo "Created profile ID: $PROFILE_ID"

# 3. Create a collector
curl -s -X POST "$API/collectors" \
  -d "replicator_ip=$REPLICATOR_IP" \
  -d 'collector_ip=10.1.10.60' \
  -d 'collector_port=2055' \
  -d 'note=API created collector' \
  -b cookies.txt -k

# 4. Assign collector to profile
curl -s -X POST "$API/profiles/$PROFILE_ID/collectors" \
  -d "replicator_ip=$REPLICATOR_IP" \
  -d 'collectors=[{"collector_ip":"10.1.10.60","collector_port":2055}]' \
  -b cookies.txt -k

# 5. Create a policy to include all exporters on port 2055
curl -s -X POST "$API/policies/$PROFILE_ID" \
  -d 'ip_range=0.0.0.0/0' \
  -d 'received_port=2055' \
  -d 'include=1' \
  -b cookies.txt -k

echo "Setup complete!"
```

4.3.3.9 Deprecated Features

The following v19 API features are no longer available in v20:

Feature	v19 Endpoint	Status in v20
Profile listening/sending ports	`/api/1/profile/add/[name]/[listeningport]/[sendingport]`	Removed - ports are on collectors/exporters
Direct exporter assignment	`/api/1/exporter/add/[ip]/[profile]`	Removed - use policies instead
Singularity toggle	`/api/1/profile/singularity/[name]/[action]`	Removed
Collector threshold	`/api/1/collector/threshold/[collector]/[threshold]`	Removed
Notate/descriptions	`/api/1/notate/[entity]/[identity]/[description]`	Removed - use `note` field on collectors or `description` on profiles
Configuration rebuild	`/api/1/rebuild`	Removed - configuration is database-driven
Show/realtime endpoints	`/api/1/show/*`	Removed
DNS check	`/api/1/dnscheck/[ip]`	Removed

4.3.4 Version upgrades

Version upgrades may include additional functionality, performance enhancements, and/or other improvements over previous versions. Fixes for certain types of issues will also be included in these updates.

4.3.4.1 Upgrading to v20.0.2

The Replicator 20.0.2 upgrade requires the target server to be running **v19.1.1** (must be provisioned with at least 2 CPU cores and 8 GB of RAM) **or higher**. For older versions, follow *this guide to upgrade to v19.1.1*.

Important

Profile data is automatically migrated when a Replicator 19.1.1 appliance is upgraded to v20.0.2. Refer to *this guide* to migrate data from a v19.1.1 appliance to a different v20.0.2 standalone or headless instance.

For v19.1.1 hardware appliance upgrades, see *this section* first.

Note

- The upgrade will take about one hour to complete.
- Due to an increase in minimum specs, older VMs with 2 GB of RAM should be provisioned with 8 GB RAM before the upgrade.
- The upgrade requires a minimum of 16 GB free space on root (/). There may be older logs (`sudo rm /var/log/messages-*`) that can be deleted to free up space.
- The primary appliance in a high-availability pair must be upgraded **before** the secondary/backup. See *this section* for instructions.

Contact *Plixer Technical Support* for assistance or clarifications.

Pre-upgrade preparation

- [Upgrades from 19.1.1] Create a backup of the v19.1.1 profile configuration data by downloading and running this utility (hardware appliances) or taking a VM snapshot (virtual appliances).
- Confirm the current password for the `replicator` SSH user (run `passwd replicator`).
- Verify that root login is disabled by running:

```
sudo sed -i 's/^#PermitRootLogin.*/PermitRootLogin no/g' /etc/ssh/sshd_config
```

- Confirm that the Replicator server has access to `https://files.plixer.com`. This check can be performed by downloading the upgrade checksum file using the following command:

```
curl -O https://files.plixer.com/plixer-repo/scrutinizer/19.7.2/replicator-install.
↪run.sha256
```

  For Replicator servers that do not have internet access, download the file from the `REPO_HOST_IP` for the *offline yum/dnf repository* instead.

Upgrading the server

Once all preparation steps have been completed, follow these steps to upgrade the appliance:

View instructions

1. SSH to the Replicator server to be upgraded as the `plixer` user (`replicator` user for upgrades from 19.1.1):

```
ssh replicator@REPLICATOR_IP
```

2. Start a new tmux session (to maintain the upgrade session if the SSH connection is lost):

```
tmux new -s upgrade
```

3. Verify that the current working directory is correct (`replicator`):

```
cd /home/replicator/
```

4. Download the Replicator 20.0.2 upgrade script and its checksum file:

```
curl -O https://files.plixer.com/plixer-repo/scrutinizer/19.7.2/replicator-install.
↪run
curl -O https://files.plixer.com/plixer-repo/scrutinizer/19.7.2/replicator-install.
↪run.sha256
```

> **Note**
>
> If the server does not have Internet access, use the `REPO_HOST_IP` for the *offline yum/dnf repository* in place of `files.plixer.com`.

5. Verify the checksum:

```
sha256sum -c replicator-install.run.sha256
```

6. Set the correct permissions for the installer:

```
chmod 755 replicator-install.run
```

7. Run the installer as the `replicator` user:

```
./replicator-install.run
```

   For offline upgrades, use:

```
REPO_HOST=REPO_HOST_IP ./replicator-install.run -- -k
```

8. After the installation script finishes running, reboot the appliance:

```
sudo shutdown -r now
```

After the reboot, the Replicator appliance will be on v20.0.2.

Offline upgrades

To upgrade a Replicator 19.1.1 server that is unable to access the default yum/dnf repository on `https://files.plixer.com/plixer-repo/scrutinizer/19.7.2`, an offline repository will need to be set up on the local network. This repository can be hosted on a Scrutinizer server or another host on the network.

To set up the offline repository on a Scrutinizer server (with IP address `REPO_HOST_IP`), follow these steps:

View instructions

1. Deploy a new Scrutinizer VM and assign the IP address REPO_HOST_IP to it.
2. Download the offline repo package and checksum file on a host with Internet access:

```
curl -O https://files.plixer.com/plixer-repo/scrutinizer/19.7.2_offline.tgz
curl -O https://files.plixer.com/plixer-repo/scrutinizer/19.7.2_offline.tgz.sha256
```

3. Start an SSH session with the Scrutinizer server as the `plixer` user:

```
ssh plixer@REPO_HOST_IP
```

4. Verify that `/var/db/big` has at least 84 GB of free disk space:

```
df -h --output='avail' /var/db/big
```

5. Create a new directory for the offline installation files and set the correct permissions to give the plixer user access to it:

```
sudo mkdir -p /var/db/big/offline
sudo chown plixer:plixer /var/db/big/offline
```

6. On the Internet-connected host, copy the offline bundle and checksum file downloaded in step 1 to the repo host:

```
scp 19.7.2_offline.tgz* plixer@REPO_HOST_IP:/var/db/big/offline/
```

7. On the Scrutinizer server, validate the checksum:

```
(cd /var/db/big/offline/ ; sha256sum -c 19.7.2_offline.tgz.sha256)
```

8. Extract the repository:

```
tar -zxvf /var/db/big/offline/19.7.2_offline.tgz -C /var/db/big/offline
```

9. Create a link to the offline repository in a directory accessible to the web server:

```
sudo -u webapp ln -sf /var/db/big/offline/plixer-repo /home/webapp/html/
```

10. Export the repo host's IP address:

```
export REPO_HOST=REPO_HOST_IP
```

Once the offline repository has been set up, follow *these steps* to proceed with the upgrade.

High-availability pairs

To upgrade a Replicator 19.1.1 high-availability (HA) pair to v20.0.2, follow these steps:

> **Note**
>
> To set up a new v20.0.0+ HA pair, follow *this guide*.

View instructions

0. [*Virtual IP pairs only*] On the primary instance, check `/etc/keepalived/keepalived.conf` and note the virtual IP address (VIP) being used (can also be obtained using `ip addr show` commands).
1. Stop and disable the `keepalived` service.

```
sudo systemctl stop keepalived
sudo systemctl disable keepalived
```

2. Verify that the VIP was released by the primary and has been taken over by the secondary instance (run on both):

```
ip a
```

3. Upgrade the primary instance following *this guide*.
4. After the upgrade, log into the web interface and verify that profiles were successfully migrated to the upgraded primary instance.
5. *Apply a new license* (must support at least two Replicator instances).
6. *Register a new instance* under **Admin** > **Resources** > **Replicators** and note the API authentication token.
7. Reboot the primary instance, and then upgrade the secondary instance following the same steps as before.
8. After the upgrade, stop and disable Scrutinizer services on the secondary instance.

```
sudo systemctl stop scrutinizer plixer_db plixer_collector plixer_webapp
sudo systemctl disable scrutinizer plixer_db plixer_collector plixer_webapp
```

9. Run the setup script:

```
sudo /usr/share/replicator/util/setup.sh
```

10. Enter the following details when prompted:
    - Configuration host (primary instance) IP address
    - API authentication token created when the new instance was created (step 6)
    - Name assigned to the new instance
11. Restart the secondary instance, and then return to **Admin** > **Resources** > **Replicators** in the web interface to verify that the secondary instance has successfully registered itself with the primary (IP address should be displayed).
12. On the same page, click on the primary instance name (usually 'Local'), and *configure high availability using the primary and secondary instances* (enter the previous VIP if necessary).
13. Click the **Save** button, and then wait two minutes.
14. Verify that `/etc/keepalived/keepalived_replicator.conf` contains the following (on both devices):

```
# this file is automatically generated do not edit
global_defs {
        script_user plixer
        enable_script_security
}
vrrp_instance VI_1 {
        state BACKUP
        interface eth0
        virtual_router_id 51
        priority 90
        advert_int 1
        preempt_delay 5
        notify  /usr/share/replicator/bin/notify.sh
        authentication {
                auth_type PASS
                auth_pass zhO2brnVF2byl4+zL/
↪ssTaCPmHFr5IMQJEIVDkJiFCCvBIttqjBptwm8c8PWbjqqN4BVieKunkVqPWIRp4fG0Q==
        }
        virtual_ipaddress {
                10.42.150.17
        }
}
```

15. Modify `/etc/keepalived/keepalived.conf` on both instances:

```
sudo bash -c 'echo "include /etc/keepalived/keepalived_replicator.conf" > /etc/
↪keepalived/keepalived.conf'
```

16. Restart the `keepalived` service on both instances:

```
sudo systemctl restart keepalived
```

Both Replicator instances should now be running v20.0.2 and set up as a high-availability pair using the same (pre-upgrade) configuration.

Hardware appliances

V19.1.1 hardware appliances that are upgraded to v20.0.2 default to the standalone role and can be used as the configuration host for headless Replicator instances.

If another appliance (standalone Replicator 20.0.2 or Plixer One/Scrutinizer 19.7.2) is going to be used as the configuration host, the hardware appliance can be converted to a headless instance instead.

Important

Profile configuration data must be backed up **before** a Replicator 19.1.1 hardware appliance is upgraded, so it can be restored after the conversion.

Upgrading and converting a v19.1.1 hardware appliance

Follow these steps to upgrade a v19.1.1 hardware appliance and convert it to a headless instance with all configuration data retained:

View instructions

1. Log in to the configuration host and *register/add a new headless instance*.
2. SSH to the source host as the `root` user and create a directory for the backup files:

```
mkdir /tmp/migration
```

3. Download the migration utility and apply the necessary permissions:

```
curl -o /tmp/migration/Replicator-19.conf-tool.sh https://files.plixer.com/plixer-
↪repo/scrutinizer/19.7.2/util/Replicator-19-conf-tool.sh
chmod 755 /tmp/migration/Replicator-19-conf-tool.sh
```

4. Run the utility to create the backup file:

```
cd /tmp/migration
./Replicator-19-conf-tool.sh backup <BACKUP_NAME>
```

5. Verify the contents of the backup file:

```
tar -vztf BACKUP_NAME.tar.gz
ls -l /home/replicator/v19_backup
```

   The files should be the same size and not be empty.

6. Copy the backup file from the v19.1.1 hardware appliance to the configuration host:

```
scp BACKUP_NAME.tar.gz plixer@CONFIGURATION_HOST_IP:/home/plixer/scrutinizer/files/
↪BACKUP_NAME.tar.gz
```

7. Follow *this guide* to upgrade the v19.1.1 hardware appliance to 20.0.2.
8. SSH to the upgraded hardware appliance as the `plixer` user:

```
ssh plixer@HARDWARE_APPLIANCE_IP
```

9. Run the setup script to demote the appliance to headless:

```
MAKE_HEADLESS=1 /usr/share/replicator/util/setup.sh
```

When prompted, provide the configuration host IP address, name assigned, and authentication token generated for the headless instance when it was registered.

10. In the configuration host web interface, navigate to **Admin** > **Resources** > **Replicators** again and verify that the hardware appliance has self-registered.
11. SSH to the configuration host as the `plixer` user and apply the necessary permissions to the utility:

```
chmod 755 /home/plixer/scrutinizer/files/Replicator-19-conf-tool.sh
```

12. Run the utility to restore the backup file to the headless hardware appliance:

```
cd /home/plixer/scrutinizer/files/
./Replicator-19-conf-tool.sh restore BACKUP_NAME.tar.gz
```

13. Navigate to the Replicator UI/page on the configuration host and verify that the profile data was successfully restored and associated with the destination headless instance.

After the backed up data has been restored, the Replicator service should be updated with the new profile configuration data and start replication within one minute.

4.3.4.2 Upgrading to v19.1.1

View guide

Because the Replicator v19.1.1 upgrade includes an OS update, v19.0.1 is **required** for the upgrade. If you are running an older version, follow *this guide to upgrade your appliance to v19.0.1*.

For assistance or clarifications, contact *Plixer Technical Support*.

Important notes

- The upgrade will take **at least** one hour to complete.
- If the Replicator appliance is able to access files.plixer.com, the `REPO_HOST` variable should be set to `files.plixer.com` for the steps outlined below. For *offline upgrades*, the IP address of the offline repo should be used instead.
- Due to an increase in minimum specs, older VMs with 2 GB of RAM should be provisioned with 4 GB RAM before the upgrade.
- The upgrade requires a minimum of 16 GB free space on root (`/`). There may be older logs (`sudo rm /var/log/messages-*`) that can be deleted to free up space.

Upgrade process

The process of upgrading a v19.0.1 Replicator appliance to v19.1.1 involves the following steps:

- Backing up the v19.0.1 database and server-specific files using the `backup` interactive mode command
- Downloading the operating system upgrade script, `olmigrate.run`, and running it a total of four times (with a reboot between runs)
- Downloading and running the Replicator v19.1.1 installation script (`replicator-install.run`)
- Verifying that the v19.0.1 data has been successfully migrated after v19.1.1 is installed

Pre-upgrade preparation

- [Hardware appliances] Create a backup of the current Replicator appliance using the backup interactive mode command (will be saved in `/home/replicator/backups`) and store it on an external system/drive.
- [Virtual appliances] Backup the current Replicator install by taking a VM snapshot.
- Confirm the current password for the `replicator` SSH user (run `passwd replicator`)
- Verify that root login is disabled by running:

```
sudo sed -i 's/^#PermitRootLogin.*/PermitRootLogin no/g' /etc/ssh/sshd_config
```

- [Offline upgrades] If the Replicator appliance does not have access to `files.plixer.com`, *set up an offline repository for this upgrade*.

Upgrading the server

Once all preparation steps have been completed, follow these steps to upgrade the appliance:

View instructions

Important

- For offline upgrades, `REPO_HOST` should point to the IP address of the *offline repo* instead of `files.plixer.com`.
- In *high-availability configurations*, complete the upgrade for the secondary appliance before the primary.
- To verify the current progress of the OS upgrade at any time:

```
cat /etc/motd
```

or check versions between runs (`NAME=` and `VERSION=` lines):

```
cat /etc/os-release
```

1. SSH to the v19.0.1 appliance to be upgraded as the `replicator` user.
2. Verify that the current working directory is correct (`replicator`):

```
cd /home/replicator/
```

3. Download the OS upgrade script and its checksum file:

```
REPO_HOST=files.plixer.com
curl -k -o olmigrate.run https://$REPO_HOST/plixer-repo/replicator/19.1.1/olmigrate.
↪run
curl -k -o olmigrate.run.sha256 https://$REPO_HOST/plixer-repo/replicator/19.1.1/
↪olmigrate.run.sha256
```

4. Validate the integrity of `olmigrate.run`:

```
cat olmigrate.run.sha256
sha256sum olmigrate.run
```

5. Set the correct permissions for the OS upgrade script:

```
chmod a+x olmigrate.run
```

6. Run the `olmigrate.run` script a total of four times:

```
REPO_HOST=files.plixer.com ./olmigrate.run -- -k
```

> **Important**
>
> Reboots between runs of the OS upgrade script (`olmigrate.run`) can take a long time. Before trying to reconnect to the appliance, start a `PING` to the Replicator IP address and wait for it to become available again. **Do NOT manually reboot the server**.

7. After the fourth `olmigrate.run` run (there will be no reboot), change directories to `/tmp` for the installation of Replicator v19.1.1:

```
cd /tmp/
```

8. Download the Replicator v19.1.1 installation script and its checksum file:

```
REPO_HOST=files.plixer.com
curl -k -o replicator-install.run https://$REPO_HOST/plixer-repo/replicator/19.1.1/
↪replicator-install.run
curl -k -o replicator-install.run.sha256 https://$REPO_HOST/plixer-repo/replicator/
↪19.1.1/replicator-install.run.sha256
```

9. Validate the integrity of `replicator-install.run`:

```
cat replicator-install.run.sha256
sha256sum replicator-install.run
```

10. Update permissions for the `replicator-install.run` script:

```
chmod a+x replicator-install.run
```

11. Run `replicator-install.run` to install Replicator v19.1.1:

```
REPO_HOST=files.plixer.com ./replicator-install.run -- -k
```

12. After the installation script finishes running, reboot the appliance:

```
sudo shutdown -r now
```

13. After the reboot, SSH to the appliance again to reset the password for the `admin` UI user:

```
manage --cli
password webui
```

After completing the above steps, the Replicator appliance will be on v19.1.1.

Offline upgrades to v19.1.1

The following instructions for setting up an offline repo are intended for upgrading to Replicator v19.1.1 only:

View instructions

1. Deploy a new Scrutinizer VM and assign an IP address to it.
2. SSH to the VM as the `plixer` user:

```
ssh plixer@SCRUTINIZER_VM_IP
```

3. Create the offline repo directory and assign it the correct permissions:

```
sudo mkdir /var/db/big/offline
sudo chown plixer:plixer /var/db/big/offline
```

4. Download the offline tar file for Replicator 19.1.1 and its checksum file:

```
curl -o /var/db/big/offline/19.1.1_offline.tgz https://files.plixer.com/plixer-repo/
↪replicator/19.1.1_offline.tgz
curl -o /var/db/big/offline/19.1.1_offline.tgz.sha256 https://files.plixer.com/
↪plixer-repo/replicator/19.1.1_offline.tgz.sha256
```

5. Validate the integrity of `19.1.1_offline.tgz`:

```
cat /var/db/big/offline/19.1.1_offline.tgz.sha256
sha256sum /var/db/big/offline/19.1.1_offline.tgz
```

6. Extract the offline tar file:

```
cd /var/db/big/offline
tar xvf 19.1.1_offline.tgz
```

7. Create a symlink in the html directory to the offline repo:

```
ln -s /var/db/big/offline/plixer-repo /home/plixer/scrutinizer/html/plixer-repo
```

After the offline repo has been set up, the VM's IP address should be used in place of `files.plixer.com` for `REPO_HOST` in the *upgrade instructions*.

4.3.4.3 Upgrading to v19.0.1

View guide

To upgrade to Replicator v19.0.1, *v18.14 or higher* is required.

> **Hint**
>
> For clarifications or assistance with upgrading, contact *Plixer Technical Support*.

Upgrade requirements

- v18.14 or higher installed
- CentOS 7 (OS)
- An active Internet connection

Upgrade instructions

Before performing an upgrade (or any other system change), it is highly recommended to use the `backup` interactive mode command to back up the Replicator database.

Hint

Backups are stored in `/home/replicator/backups`. To restore from a backup file, use the restore command.

Important

When upgrading Replicator appliances in HA configurations, the fail-over appliance should be upgraded before the primary to minimize downtime.

The following instructions cover the upgrade process for both primary and fail-over appliances in HA configurations:

View instructions

1. SSH to the appliance as the `replicator` user and start a new tmux session:

```
tmux new -s upgrade
```

2. Download the installer/upgrade script:

```
cd /tmp
curl -o replicator-install.run https://files.plixer.com/plixer-repo/replicator/19.0.
↪1/replicator-install.run
```

3. Download the checksum file and validate the integrity of the `replicator-install.run` file:

```
curl -o replicator-checksums.txt https://files.plixer.com/plixer-repo/replicator/19.
↪0.1/replicator-checksums.txt
cat replicator-checksums.txt
sha256sum replicator-install.run
```

4. Set the correct permissions for the installer:

```
chmod 755 replicator-install.run
```

5. Run `replicator-install.run`:

```
./replicator-install.run
```

After the upgrade is complete, the appliance will automatically reboot. To verify that the upgrade was successful, launch interactive mode and check the version number when the tool loads.

Note

After upgrading a fail-over appliance in an HA deployment, use the `role test secondary` interactive mode command to verify the current HA configuration before proceeding to upgrade the primary appliance.

4.3.4.4 Upgrading to v18.14

View guide

To upgrade to Replicator v18.14 from v18.5 or higher, follow the steps described below.

> **Hint**
>
> For clarifications or assistance with upgrading, contact *Plixer Technical Support*.

Upgrade requirements

- v18.5 or higher installed
- An active Internet connection

Upgrade instructions

Before performing an upgrade (or any other system change), it is highly recommended to use the `backup` interactive mode command to back up the Replicator database.

View instructions

1. SSH to the appliance as the `replicator` user and start a new tmux session:

```
tmux new -s upgrade
```

2. Download the upgrade script:

```
cd /home/replicator/files
curl -k -o upgrade_18.14.sh https://files.plixer.com/downloads/replicator/18/
↪upgrade_18.14.sh
```

3. Set the correct permissions for the upgrade script:

```
chmod 755 upgrade_18.14.sh
```

4. Run `upgrade_18.14.sh`:

```
./upgrade_18.14.sh
```

After the upgrade is complete, the appliance will automatically reboot. To verify that the upgrade was successful, launch interactive mode and check the version number when the tool loads.

4.3.5 Profile migration

Profile configuration data from a Replicator 19.1.1 server can be migrated to a different Replicator 20.0.0+ appliance/instance using the migration utility included with standalone Replicator appliances and Plixer One/Scrutinizer deployments.

> **Note**
>
> Profile data is automatically migrated when a Replicator deployment is upgraded from v19.1.1 to v20.0.0+.

The following Replicator 20.0.0+ deployment types are supported as migration destinations:

- Standalone appliances
- Headless appliances
- Local instances on Scrutinizer

4.3.5.1 Migration to a standalone Replicator instance

The following instructions cover profile data migration from a Replicator 19.1.1 server to a standalone Replicator 20.0.2+ instance (or a local instance on Scrutinizer 19.7.2+).

View guide

Requirements

- Source host - Replicator 19.1.1 appliance
- Destination host - Fully deployed and licensed standalone Replicator 20.0.2+ deployment (or local instance on Scrutinizer 19.7.2+)
- Temporary host - A temporary location to use for moving files between the source and destination hosts
- Migration utility (`Replicator-19-conf-tool.sh`) - Download from `https://files.plixer.com/plixer-repo/scrutinizer/19.7.2/util/Replicator-19-conf-tool.sh` or copy from `/home/plixer/scrutinizer/files/` on the destination host

Important

Profile data can only be migrated between source and destination hosts with the same IP address.

Migration procedure

1. SSH to the source host as the `root` user and create a directory for the migration files:

```
mkdir /tmp/migration
```

2. Download the migration utility and apply the necessary permissions:

```
curl -o /tmp/migration/Replicator-19.conf-tool.sh https://files.plixer.com/plixer-
↪repo/scrutinizer/19.7.2/util/Replicator-19-conf-tool.sh
chmod 755 /tmp/migration/Replicator-19-conf-tool.sh
```

Note

If the source host cannot access the Internet, the utility can be copied from the destination host via the temporary host.

3. Run the utility to create the backup file:

```
cd /tmp/migration
./Replicator-19-conf-tool.sh backup <BACKUP_NAME>
```

This will create the profile data backup file `BACKUP_NAME.tar.gz`.

4. Verify the contents of the backup file:

```
tar -vztf BACKUP_NAME.tar.gz
ls -l /home/replicator/v19_backup
```

The files should be the same size and not be empty.

5. Copy the backup file from the source host to the temporary host, and then shut down the source host:

```
scp BACKUP_NAME.tar.gz USER@TEMP_HOST_IP:/DESTINATION_DIR/BACKUP_NAME.tar.gz
shutdown -p now
```

6. After the source host has been completely shut down, *deploy the destination host* and assign it the same IP address as the source host.

7. Copy the backup file from the temporary host to the destination host:

```
scp /DESTINATION_DIR/BACKUP_NAME.tar.gz plixer@DESTINATION_HOST_IP:/home/plixer/
↪scrutinizer/files/BACKUP_NAME.tar.gz
```

8. SSH to the destination host as the `plixer` user and apply the correct permissions to the utility:

```
chmod 755 /home/plixer/scrutinizer/files/Replicator-19-conf-tool.sh
```

9. Run the utility to restore the backup file on the destination host:

```
cd /home/plixer/scrutinizer/files/
./Replicator-19-conf-tool.sh restore BACKUP_NAME.tar.gz
```

10. Navigate to the Replicator UI on the destination host and verify that the profile data was successfully restored.

After the migration has been completed, the Replicator service should be updated with the new profile configuration data and start replication within one minute.

4.3.5.2 Migration to a headless Replicator instance

The following instructions cover profile data migration from a v19.1.1 deployment to a headless Replicator 20.0.2+ instance.

View guide

Requirements

- Source host - v19.1.1 Replicator appliance
- Destination host - Headless Replicator 20.0.0+ instance *registered with the configuration host and deployed*
- Configuration host - Fully deployed and licensed standalone Replicator 20.0.0+ deployment with at least an evaluation license; license must support at least two instances (configuration host and destination headless instance)
- Migration utility (`Replicator-19-conf-tool.sh`) - Download from `https://files.plixer.com/plixer-repo/scrutinizer/19.7.2/util/Replicator-19-conf-tool.sh` or copy from `/home/plixer/scrutinizer/files/` on the configuration host

Important

The destination host must be deployed with the same IP address as the source host **after** the backup file has been created and the source host has been shut down.

If the destination host has already been deployed, *update its IP address* to the correct one after the source host has been shut down but before running the migration utility (step 9 below).

Migration procedure

1. SSH to the source host as the `root` user and create a directory for the migration files:

```
mkdir /tmp/migration
```

2. Download the migration utility and apply the necessary permissions:

```
curl -o /tmp/migration/Replicator-19.conf-tool.sh https://files.plixer.com/plixer-
↪repo/scrutinizer/19.7.2/util/Replicator-19-conf-tool.sh
chmod 755 /tmp/migration/Replicator-19-conf-tool.sh
```

> **Note**
>
> If the source host cannot access the Internet, copy the utility from the configuration host:
>
> ```
> scp /home/plixer/scrutinizer/files/Replicator-19-conf-tool.sh replicator@SOURCE_
> ↪HOST_IP:/tmp/migration/
> ```

3. Run the utility to create the backup file:

```
cd /tmp/migration
./Replicator-19-conf-tool.sh backup <BACKUP_NAME>
```

   This will create the profile data backup file `BACKUP_NAME.tar.gz`.

4. Verify the contents of the backup file:

```
tar -vztf BACKUP_NAME.tar.gz
ls -l /home/replicator/v19_backup
```

   The files should be the same size and not be empty.

5. Copy the backup file from the source host to the configuration host, and then shut down the source host:

```
scp BACKUP_NAME.tar.gz plixer@CONFIGURATION_HOST_IP:/home/plixer/scrutinizer/files/
↪BACKUP_NAME.tar.gz
shutdown -p now
```

6. *Deploy the destination host appliance* and assign it the same IP address previously used by the source host.
7. Navigate to **Admin** > **Resources** > **Replicators** again and verify that the destination Replicator instance has self-registered and has the correct IP address.
8. SSH to the configuration host as the `plixer` user and apply the necessary permissions to the utility:

```
chmod 755 /home/plixer/scrutinizer/files/Replicator-19-conf-tool.sh
```

9. Run the utility to restore the backup file to the destination host:

```
cd /home/plixer/scrutinizer/files/
./Replicator-19-conf-tool.sh restore BACKUP_NAME.tar.gz
```

10. Navigate to the Replicator UI/page on the configuration host and verify that the profile data was successfully restored and associated with the destination headless instance.

After the migration has been completed, the Replicator service should be updated with the new profile configuration data and start replication within one minute.

Hint

Profile data from multiple 19.1.1 Replicator instances can be migrated to new headless Replicator 20.0.0+ appliances by repeating the steps above and deploying each destination host with an IP address matching a source.

4.4 Additional Resources

FAQ

Frequently asked questions

FAQ **Changelog**

Replicator updates and version history

Replicator changelogs **Glossary**

Glossary of terms used in Replicator

Glossary **Attributions**

Open source and third-party licenses

Third-party attributions **Plixer technical support**

Plixer Technical Support is available with an active maintenance contract. Contact our support team at:

- **Phone:** +1 (207) 324-8805 ext 4
- **Website:** https://www.plixer.com/support/

4.4.1 FAQ

Note

For additional questions or concerns, contact *Plixer Technical Support*.

Yes. For instructions on how to configure a secondary or backup Replicator appliance see the section on high availability configurations.

A firewall or access control list may be blocking traffic to the Replicator appliance. To verify that it can see traffic from a device, run `snoop [NETWORK_DEVICE_IP]` from the REPLICATOR> prompt.

Each Replicator appliance currently only supports a single adminstrator account for the web interface. However, future updates to the product may add support for multiple local user accounts and roles.

When a new exporter starts sending UDP packets to the Replicator appliance, it may take up to two minutes for the packets to be received by collectors.

To view all currently unassigned exporters, select *Exporters Not in a Profile* in the **View** dropdown menu in the **Exporters** tab of the web interface or run `exporters noprofile` from the REPLICATOR> prompt.

The default behavior for the Replicator appliance is to drop all packets that come into an interface that the host has no route to. To change this, find the `net.ipv4.all.rp_filter` setting in the `/etc/sysctl.conf` file and change its value to `0`.

If you are using Scrutinizer's distributed architecture to handle an extremely large number of flows and/or exporters, you can enable the *Auto Replicate* feature to have it manage Replicator profiles for its collectors and automatically assign exporter streams to them as they're added. Additional information and instructions on how to set up Replicator integration can be found in the Scrutinizer documentation here.

Yes, IPFIX is supported.

By default, the Replicator appliance replicates syslog messsages received from exporters and forwards them to their assigned collectors in the same format. To have syslogs automatically converted to IPFIX before being forwarded, run `setting enable convertSyslog` from the REPLICATOR> prompt.

To send the appliance's syslog notifications and/or IPFIX metrics to multiple collectors, first configure the Replicator appliance to send the packets back to itself. After that, create a profile with the same appliance as an exporter and assign collectors as needed.

Note

Since there are separate settings for syslog notifications and IPFIX metrics, they will require separate profiles.

A loop is created when a profile is configured in a way that incoming packets will be sent back to the source. This can happen when a collector is added to a profile with an inclusion policy that defines the same IP address as an exporter.

Yes. The Replicator will automatically verify updates to a profile's settings to ensure that no loops are created when new exporters and/or collectors are added.

To change the root password, SSH to the appliance as the `root` user and issue the `passwd` command.

To change the password for the web interface `admin` user, SSH to the appliance as the `replicator` user and run `password webui` from the REPLICATOR> prompt.

To change the hostname and IP address of the appliance, log in as the root user, and then run `/home/replicator/conf/sethostname.sh`. Alternatively, `system change` can be run from the REPLICATOR> prompt.

The polling interval is controlled by the *Update Interval* setting on the **Settings** tab/page of the web interface and can be set anywhere between 30 to 60 seconds.

The secondary appliance checks the primary for changes every 5 minutes. If changes are detected, the secondary appliance's configuration is updated.

Jumbo frame sizes up to 65534 bytes are supported by the Replicator appliance. To take advantage of this, interfaces must be configured to support the maximum packet size expected for replication. Fragmented packets are also supported.

4.4.2 Replicator changelogs

Changelog entries are displayed in the format **DESCRIPTION (Ticket Number)**.

Note

- For more information on Replicator, visit www.plixer.com or contact *Plixer Technical Support*.
- Please refer to our End of Life Policy for EOL schedule details.

4.4.2.1 Replicator v20.0.2 - January 2026

Changelog

Fixes

- Replicator configuration migration version checking (243)
- Replicator hardware upgrade failing from v19.1.1 > v20.0.1 (244)
- Replicator UI freezes upon loading of the topology view with a high exporter count (248)

4.4.2.2 Replicator v20.0.1 - November 2025

Changelog

Fixes

- Addressed various security issues
- Headless replicator failing to integrate when created with space characters in name (233)
- Replicator upgrade fails (234)

4.4.2.3 Replicator v20.0.0 - October 2025

> **Note**
>
> As of v20.0.0, Replicator is fully integrated into the Plixer One platform UI. See *this section* for details on added features and functionality, or visit the Scrutinizer online manual for further information.

Changelog

New features

- Add Support for IPv6
- HA dual exporter profile type
- Manage Auto Replicate through the UI
- Single Sign On (SAML)
- Support giant packets / Fragmented packets

Enhancements

- Auto Replicate: Ability to support only new device discovery
- Auto Replicate: Add logic so that if a policy is modified in the Seed profile, that policy change is applied in any collector profiles that include the same policy.

Fixes

- Issues with more than 103 collectors (19)
- User configurable timezones (8)

4.4.2.4 Replicator v19.1.1 - October 2024

> **Note**
>
> This release addresses CentOS going EOL. To migrate the OS to Oracle Linux 9, the Replicator must be on version 19.0.1. Please contact *Plixer Technical Support* with any questions.

Changelog

New features

- KVM virtual appliance image

Fixes

- Addressed various security issues
- System metrics errors (167)
- HA configuration errors (170)
- Set license command error (174)
- net-snmp package not in repo (178)
- sethostname.sh script missing (187)
- Set password from CLI (189)
- Web certificate file name (193)

4.4.2.5 Replicator v19.1.0 - June 2024

Changelog

> **Note**
>
> This release addresses CentOS going EOL. To migrate the OS to Oracle Linux 9, the Replicator must be on version 19.0.1. Please contact *Plixer Technical Support* with any questions.

New features

- Oracle Linux v9.4

Fixes

- Addressed various security issues
- Added back missing enable_ssl.sh script (161)

4.4.2.6 Replicator v19.0.1 - January 2024

Changelog

Fixes

- Addressed various security issues
- Fixed authentication issue (150)

4.4.2.7 Replicator v19.0.0 - November 2023

Changelog

Enhancements

- Improved UI responsiveness
- Updated EULA
- Updated branding

Fixes

- Addressed various security issues
- API may hang on invalid input (23)
- Exporter id out of range (25)
- Remote root access disabled by default (34)
- Improved process management (60)

4.4.2.8 Replicator v18.14.1 - January 28, 2020

Changelog

New features

- Updated the EULA
- New Replicator UI skin

Fixes

- Fixed a potential issue when setting up High Availability pairs (931)
- Replicator's vitals processes can no longer enter a bad state, which may cause it to lock up indefinitely (1739)
- Replicator excludes Collectors from being Exporters in Profiles with the same destination port the Collector is already receiving on (1771)
- Exporters will now be added to Profiles correctly regardless of the time of the matching Policy's creation (1891)
- Very high volume Replicators will no longer crash due to a database integer overflow (2007)

4.4.2.9 Replicator v18.12.14 - January 25, 2019

Changelog

Enhancements

- Future Replicator releases will no longer support CentOS 6

Fixes

- Replication no longer starts and restarts spontaneously (410)
- Replicator will no longer attempt to sync with its own database (428)
- `system change` now works on all configurations (436)
- Replicator stats are now more reliable (611)

4.4.2.10 Replicator v18.5 - May 31, 2018

Changelog

Enhancements

- Updated the EULA (5633)

Fixes

- `show config` output now has profile names enclosed in quotation marks (“”) (25257)
- Fixed an issue where Apache fails to start if SSL is enabled/setup via install script or enable_SSL.sh (25292)
- Updated licensing checks (25653)
- Fixed an issue where an install script was pointing to a previous version (25770)
- Added the new online manual from docs.plixer.com (25828)
- Fixed an issue where `policy remove profile include/exclude` results in an internal error (500) but still removes profile (25832)
- Fixed an issue where the refresh countdown timer would default to one (1) day (26000)

4.4.2.11 Replicator v18.1 - January 30, 2018

Changelog

New features

- LDAP authentication support
- Profiles displayed in alphabetical order
- Replicator install and upgrade logs
- Backup process from the CLI
- Ability to make API calls via HTTPS
- Support for upper case letters and spaces in profile names
- Ability to restrict snoop command by port
- Current Plixer Replicator version displayed under the Status LED
- Postgres replicator database support

Fixes

- Improved performance and responsiveness of the web interface (22918)
- Replicator now replicates SNMP traps on low port numbers (23820)
- Exporters no longer enter a false Alarm state at start up (23824)
- Exporters not in Profiles and not sending packets are no longer displayed under ‘Exporters Not in Profiles’ indefinitely (23828)
- Search filters are no longer lost on page refresh (23904)
- Fixed an issue that could result in phantom Collector alarms (23977)
- Upgrades no longer reset the web interface password (24299)

4.4.2.12 Replicator v17.6 - July 14, 2017

Changelog

New features

- Web interface updates
- Fully supported and documented API
- Ability to receive packets from multiple interfaces
- Profile singularity

Fixes

- Licensing says expired one day before expiration date (19835)
- ICMP drops have been added to iptables (19396)
- Semicolon at end of command yields unexpected results (19285)
- Removing a Policy doesn't remove the associated Exporters (20848)
- Replicator falsely reporting more packets inbound than out (21086)

4.4.2.13 Replicator v16.9 - October 3, 2016

Changelog

New features

- New web interface
- Fully supported and documented API
- Ability to receive packets from multiple interfaces
- Profile singularity

Fixes

- Licensing says expired one day before expiration date (19835)
- ICMP drops have been added to iptables (19396)
- Semicolon at end of command yields unexpected results (19285)
- Removing a policy doesn't remove the associated Exporters (20848)
- Replicator falsely reporting more packets inbound than out (21086)

4.4.3 Glossary

This glossary is a reference for terms and concepts used in the Replicator software environment or this product manual.

4.4.3.1 Replicator

View content

Alarm Policy
: Rule sets that define what types of network behavior or activity should be monitored as events and trigger alarms

Collectors
: SIEMs, flow collectors, SNMP trap receivers, and other network management systems that capture, analyze, and report on flow data sent by exporters

EULA (End-User License Agreement)
A legal agreement between Replicator and the user, outlining the terms and conditions, including usage rights, restrictions, and liability limitations

Events
Changes in an endpoint's state or behavior that may result in profile reassignment and can be used to draw attention to endpoints of interest

Exporters
Network devices, such as routers, switches, or servers that can send traffic/activity logs as flows to external systems, such as Replicator and Scrutinizer

Policy
A subnet/CIDR-based rule that automatically includes or excludes matching exporters in a profile

Profile
A user-defined replication configuration that defines the packet streams (based on exporters and in/listening ports) that should be routed to one or more collectors

4.4.3.2 General networking

View content

2LD (Second-level Domain)
Part of the naming convention for domain names. For example, in example.com, *example* is the second-level domain of the .com TLD (Top level domain)

3LD (Third-level Domain)
For example, in www.mydomain.com, *www* is the third-level domain

ACK (Acknowledgment Code)
A unique signal sent by a computer to show that it has successfully transmitted data

ACL (Access Control List)
A set of rules governing access to a particular object or system resource

Active Directory / AD
Proprietary directory service offered by Microsoft, which allows for centralized management of users, devices, and other IT assets

API (Application Programming Interface)
A software component that allows applications to share data and functionality

ARP (Address Resolution Protocol)
Protocol that maps a dynamic IP address to a physical machine's permanent MAC address in a local area network (LAN)

CA (Certification Authority)
A trusted entity that issues, signs, and stores digital certificates

CDP (Cisco Discovery Protocol)
Protocol used by Cisco devices to allow neighboring networking devices to learn about each other

CIDR (Classless Inter-Domain Routing)
An IP addressing method that improves the efficiency of allocating IP addresses

CLI (Command-line Interface)
A text-based interface for applications and operating systems that allows a user to enter commands

Collector
SIEMs, Flow Collectors, SNMPTrap Receivers, or other network management systems that analyze data forwarded from networked devices

DHCP (Dynamic Host Configuration Protocol)
Network management protocol used to automatically assign IP addresses and other communication parameters to devices on an Internet protocol network

DNS (Domain Name System)
A system by which computers and other devices on the Internet or Internet protocol networks are uniquely identified using names matched to their IP addresses

Egress
Traffic that exits a device or network

Endpoint
An entity (device, service, node, etc.) at the end of a network communication channel

Encapsulated Remote SPAN (ERSPAN)
Encapsulates mirrored traffic in GRE (Generic Routing Encapsulation) and sends it over Layer 3 networks

ESX (Elastic Sky X)
A pre-configured, ready-to-deploy virtual machine (VM) designed to run on VMware ESX or ESXi

Exporter
A networked device such as a router, switch, or server that generates data and sends it to the flow collector device

Fault tolerance
A system's ability to continue operating without interruptions in the event of hardware or software failure

FQDN (Fully Qualified Domain Name)
The complete address of a computer, host, or any other entity on the Internet

GRE (Generic Routing Encapsulation)
A tunneling protocol developed by Cisco Systems

Hyper-V
A pre-configured, ready-to-deploy virtual machine designed to run on Microsoft Hyper-V, typically packaged in VHD/VHDX format

ICMP (Internet Control Message Protocol)
A protocol used for devices within the network to determine possible network issues

Identity Provider (IdP)
A third-party entity and/or service that stores and manages identities and credentials for use by other websites, applications, or other digital resources

IP address
A unique numerical label assigned to a networked device

IPFIX (Internet Protocol Flow Information Export)
A protocol intended to collect and analyze the flow data from supported network devices

KVM (Kernel-based Virtual Machine)
A pre-configured virtual machine designed to run on KVM hypervisors, packaged in formats like QCOW2 or OVA for easy deployment in Linux-based virtualization environments

Latency
The latency of a network is the time it takes for a data packet to be transferred from its source to the destination

LDAP (Lightweight Directory Access Protocol)
An open, cross-platform protocol used to access and maintain directory services for assets in an Internet protocol network

LLDP (Link Layer Discovery Protocol)
A vendor-neutral protocol used by devices on IEEE 802 networks to advertise their identity, capabilities, and other information

MAC (Media Access Control) address
A unique hardware identifier typically assigned by manufacturers to network adapters and devices

MIB (Management Information Base)
A database that stores information used for managing a network

MTTR (Mean Time to Resolution)
The average amount of time between the detection and remediation of a security threat or incident

NDR (Network Detection and Response)
A cybersecurity solution that use machine learning to detect cyber threats and aid remediation

Network interface
A (physical or software-based) point of connection between a network entity and the rest of the network

NIC (Network Interface Card)
Adapter that provides devices network connections, either wired or wireless

NID (Network Infrastructure Device)
Any device, such as an access point, router, or switch, that provide the means for entities to communicate with each other over a network

NTP (Network Time Protocol)
A networking protocol used to synchronize device clocks over the Internet

NXDOMAIN (No Existing Domain)
An error message that means that a domain mentioned in the Domain Name System (DNS) query does not exist

Open port
A TCP or UDP port that has been configured to accept packets

OUI (Organizationally Unique Identifier)
A unique 24-bit number in a MAC address that identifies the vendor or the manufacturer of the device

OVF (Open Virtualization Format)
An open source standard for packaging and distributing virtual machines and software applications

Packet
A block of data transmitted across a network

PDU (Protocol Data Unit)
An individual unit of information exchanged by entities on a network using the same protocol

PostgreSQL
An open-source relational database management system (RDBMS) that supports both SQL and JSON querying

PXE (Preboot Execution Environment)
A network booting protocol that allows computers to boot from a network rather than a local storage device like a hard drive or USB

RADIUS (Remote Authentication Dial-In User Service)
A client-server AAA (authentication, authorization, accounting) protocol used to manage remote user access to a network

Redundancy
The state of having duplicate or alternative services as backups to allow for continuous availability

REST API (Representational State Transfer Application Programming Interface)
A set of rules that allows systems to communicate over the web using standard HTTP methods

Router
A device that forwards or routes data packets to devices on a network

Server
A system or device that provides resources, data, services, or applications to other devices over a network

Single Sign-On (SSO)
Allows the integration of third-party authentication services for user access to the Replicator web interface

SIP/RTP (Session Initiation Protocol/Real Time Protocol)
SIP is the control protocol, and RTP is the payload protocol used to send and receive Voice over IP (VoIP)

SNMP (Simple Network Management Protocol)
An IP network protocol used to collect data related to state and/or behavior from devices on a network

SNMP trap
An alert message that is initiated by an SNMP-enabled device to notify the management system of significant events or changes in status

Software agent
A persistent piece of software that performs certain actions and/or interacts with its environment on behalf of a user or another program

SPAN (Switched Port Analyzer)
A dedicated port on a switch that takes a mirrored copy of network traffic from within the switch to be sent to a destination

SSDP (Simple Service Discovery Protocol)
A network protocol used for advertising and discovering network services

SSH (Secure Shell Protocol)
A network communication protocol that allows network services to be used securely over an unsecured network

SSL (Secure Sockets Layer)
A protocol for establishing secure connections between networked devices

STIX (Structured Threat Information eXchange)
An industry-standard file format for the exchange of threat information between organizations and platforms

Suricata
A network threat detection engine used to analyze network traffic and identify potential security threats

Switch
A device that connects devices in a network and allows them to communicate with each other

SYN scan
A port scanning technique that allows for the discovery of the status of a communications port without establishing a full connection

Syslog
A cross-platform network logging protocol used to send and/or receive alerts between different devices on a network

TACACS+ (Terminal Access Controller Access-Control System)
A protocol where the remote access server and the authentication server provide validation for users attempting to access the network

TAXII (Trusted Automated eXchange of Indicator Information)
A protocol that allows the transmission of threat information, primarily in STIX format, between systems and organizations

TCP (Transmission Control Protocol)
A connection-oriented protocol that enables the bidirectional exchange of messages between devices on the same network

TLS handshake
The process that starts secure communication between a client and a server

TSIG (Transaction Signature)
A protocol that secures DNS packets and allows a Domain Name System to authenticate updates to the DNS database

TTL (Time To Live)
A field in the IP packet header that specifies the maximum number of hops (or router passes) a packet can take before being discarded

UDP (User Datagram Protocol)
A communication protocol for transmitting messages between applications and programs in a network

Virtual appliance
A pre-configured virtual machine image with pre-installed software that is meant to serve a specific function

VoIP (Voice over Internet Protocol)
A technology that allows voice calls using an internet connection

VPC (Virtual Private Cloud)
A secure and private cloud hosted in a public cloud

VRF (Virtual Routing and Forwarding)
A technology that separates routing tables to isolate management traffic to the management interface

Web server banner
A text-based greeting message, which includes information like open ports, services, and version numbers, returned by a web host

4.4.4 Third-party attributions

Certain open source or other third-party software components are integrated and/or redistributed with Replicator software. The licenses are reproduced here in accordance with their licensing terms.

These terms only apply to the libraries themselves, not Replicator software.

4.4.4.1 Apache 2.0 License

- **Hogan.js** (https://github.com/twitter/hogan.js/blob/master/LICENSE) - Copyright (c) 2011 Twitter, Inc.

4.4.4.2 BSD 3-Clause License

- **D3.js** (https://github.com/d3/d3/blob/master/LICENSE) - Copyright (c) 2010-2014 2010-2017 Mike Bostoc
- **jsSHA** (https://github.com/Caligatio/jsSHA/blob/master/LICENSE) - Copyright (c) 2008-2017 Brian Turek

4.4.4.3 GNU GPL 2.0

- **UDP Samplicator** (https://github.com/sleinen/samplicator/blob/master/COPYING) - Copyright (c) 2000-2015 Simon Leinen

4.4.4.4 MIT License

- **Backbone.js** (https://github.com/jashkenas/backbone/blob/master/LICENSE) - Copyright (c) 2010-2017 Jeremy Ashkenas, DocumentCloud
- **C3.js** (https://github.com/c3js/c3/blob/master/LICENSE) - Copyright (c) 2013 Masayuki Tanaka

- **JQuery** (https://jquery.org/license/) - Copyright jQuery Foundation and other contributors, https://jquery.org This software consists of voluntary contributions made by many individuals. For exact contribution history, see the revision history available at https://github.com/jquery/jquery
- **JQuery.floatThread.js** (https://github.com/mkoryak/floatThead/blob/master/LICENSE) - Copyright (c) 2012-2017 Misha Koryak
- **JustGage** (https://github.com/toorshia/justgage/blob/master/LICENSE) - Copyright (c) 2012-2015 Bojan Djuricic
- **Raphaël** (https://github.com/DmitryBaranovskiy/raphael/blob/master/license.txt) - Copyright © 2008-2013 Dmitry Baranovskiy, Sencha Labs
- **Underscore.js** (https://github.com/jashkenas/underscore/blob/master/LICENSE) - Copyright (c) 2009-2017 Jeremy Ashkenas, DocumentCloud and Investigative Reporters & Editors