anodot

We monitor your business.

On Premise Installation Guide Version 3.2

November 2020

Introduction

Anodot's anomaly detection solution provides a single, scalable platform for visualizing, analyzing, and searching multiple time-series metrics in real-time.

Using patented machine learning algorithms, Anodot rapidly finds and creates alerts for data patterns that deviate from expected norms.

By identifying and grouping these anomalies, and then classifying them by severity and other factors, Anodot empowers you to investigate the root causes of anomalies, to pinpoint performance issues, and to focus on business opportunities.

Start (UTC): May 11, 21:01
Score: 22
Duration: 1 day
Season: weekly

37f	33	ML
Package type	Price	Region

Anodot's Autonomous Analytics is provided either as a full SaaS product or as an on-premise managed service. The on-premise version contains several layers of security that prevent unauthorized data access and demonstrate accountability for actions taken.

This on-premise approach enables customers to overcome specific security and regulatory challenges while maintaining the functionality and quality of the SaaS product. This guide describes:

- Administration and Responsibilities
- On-Premise Installation Requirements
- Network and Connectivity Requirements
- DNS and SMTP Setup
- Firewall Requirements
- Managed Service Requirements
- Hardware Requirements
- Software Requirements
- Load Balancer Setup
- Backup and Restore

Administration and Responsibilities

Installation tasks and responsibilities are allocated to Anodot and its customers as follows:

Task	Responsibility
Purchasing and maintaining hardware	Customer
Initial installation and upgrades	Anodot: On-site installation by Anodot engineers.
Initial setup and configuration	Anodot: DNS setup, email setup, account creation and admin user registration.
System and data center monitoring	Customer: including host health, network health, storage etc.
Application health and monitoring	Anodot: Anodot tracks application metrics and anomalies continuously on a remote data center. It notifies the customer as soon as any issue is identified and initiates a resolution process.
Admin remote access/troubleshooting	Customer: Must provide VPN SSH access to the cluster for troubleshooting purposes.
Backup maintenance	Customer: Anodot will store backup files in a specified and agreed location in the cluster.

On-Premise Installation Requirements

Scaling and High Availability

Metrics are a unique combination of dimensions and measure values.
Anodot provides for the processing of up to 250k Metrics per node. For high availability purposes, each node must run on a separate physical machine.

Scaling Matrix Example

Number of Metrics	Samples per Second	Number of Nodes	High Availability?
250K	5K	1	No
500K	10K	3	Yes
750K	15K	4	Yes
1M	23K	6	Yes
5M	120K	20	Yes
10M	230K	38	Yes

Network and Connectivity Requirements

A fast and reliable network is important for distributed system performance. Lower latency helps ease communication between nodes, while high bandwidth helps shared movement and recovery. Modern data center networking (1 GbE) is enough for most clusters.
A private subnet with latency of less than 1ms between hosts is required to host the Worker nodes.
A public and private subnet is required for the Management host (accessed via VPN).
Avoid clusters that span multiple data centers, even if these data centers are close to one another. Avoid clusters that span large geographic distances.

Network Members

- **Kubernetes cluster**: Cluster comprising Controller-Manager, API Server and Scheduler installed on top of the infrastructure network. Communicates by internal router and DNS.
- **Controller-Manager host** (bastion/jump host): Host with specific access permissions with VPN and SSH connections allowed from a specific IP for Anodot engineers, who do initial cluster installations along with the Anodot application, plus provide additional assistance with configuration and support.
  - **Worker VMs**: Hosts that serve clusters installed on Kubernetes and the application, plus dependencies further on. They should be in the same subnet to enable communication between each other during cluster setup and normal cluster work. Workers don't require

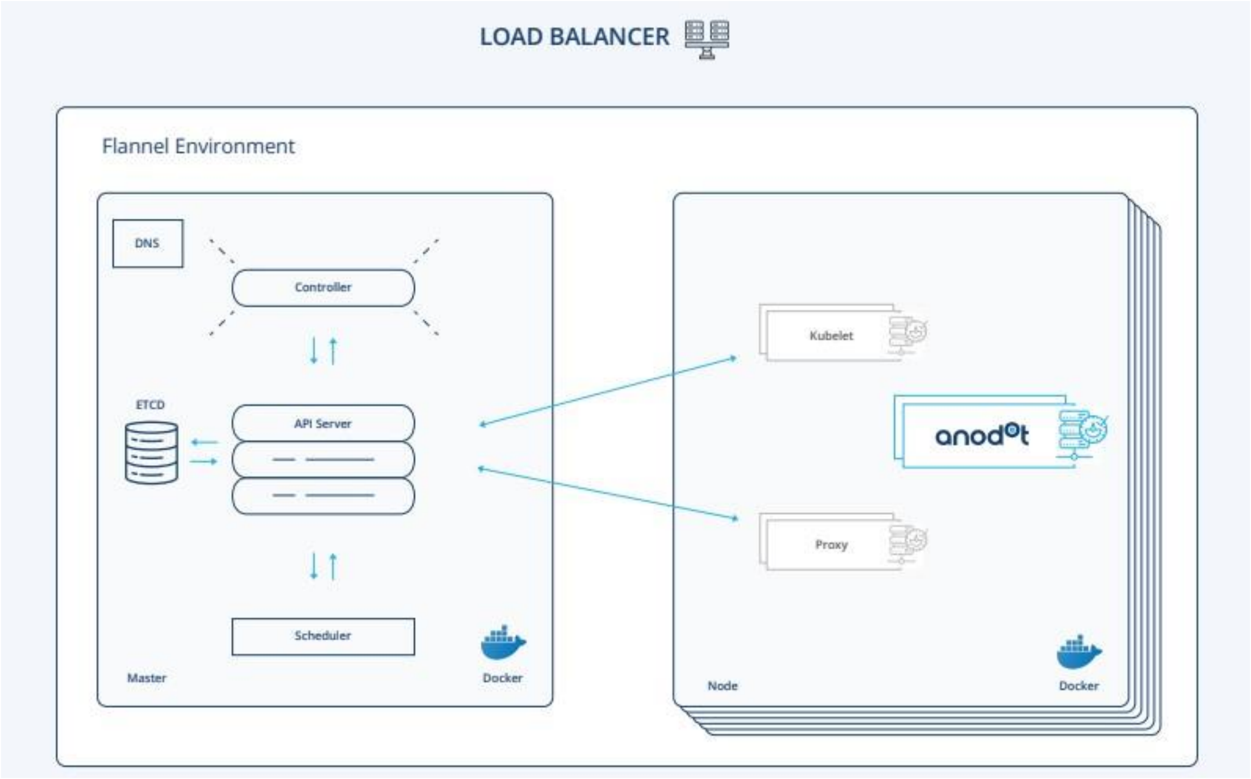
public network access, only internal communication with the Manager host, database, and SMTP host for email notifications and alerts.

- **Load balancer**: Load balancer listener over 443 to ephemeral ports of all worker nodes.

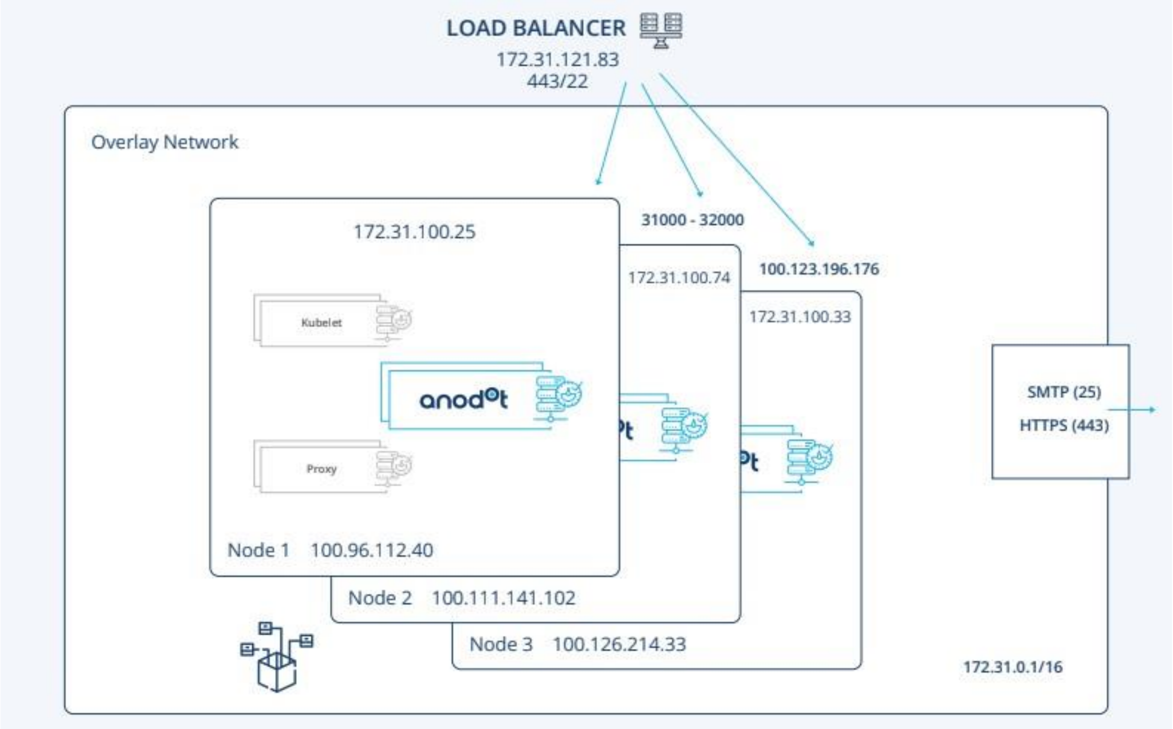
Installation Outline

Anodot’s application runs on top of the Kubernetes framework. Kubespray provides a simple way to install multi-master High Availability clusters and deploy business-critical applications.

Architecture



Deployment Example



DNS and SMTP Setup

Initial setup and configuration, including SMTP server and port, account creation, and admin user registration is done by Anodot engineers. Anodot requires:

- A resolvable DNS host record for every user in the organization that requires access.
- Credential settings and the connection string to the SMTP relay.

SMTP Required Properties

- SSL connection enabled - true/false
- Authenticated SMTP - true/false
- SMTP Port, for example, 465,25
- SMTP Host - either DNS or IP address
- Authenticated SMTP user - <xxx>
- Authenticated SMTP password - <xxx>

Firewall Requirements

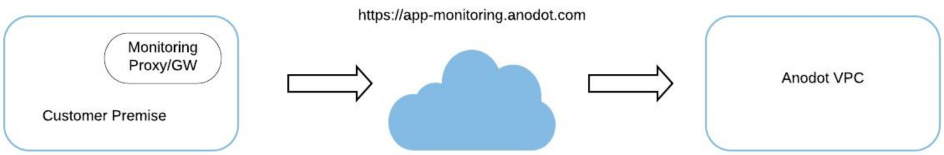
Direction	Ports	Description
Inbound + Outbound private network	SSH (22)	For Managed Service Support by SSH connection
Inbound + Outbound private network	HTTP (80,8080,443)	Internal traffic / Rest API
Inbound + Outbound private network	TCP 31000-32000	Ephemeral ports to be used for internal cluster communications
Inbound + Outbound private network	TCP 9042	Cassandra - CQL native transport port
Inbound + Outbound private network	TCP 9200, 9300, 9400	Elasticsearch communications
Inbound + Outbound private network	TCP 27017	MongoDB - The default port for mongod and mongos instances
Inbound + Outbound private network	TCP 2181, 9092	Zookeeper client port, Kafka TCP port
Inbound + Outbound public network	TCP 25 / 587	SMTP outgoing port
Inbound + Outbound public network	TCP 443	Webhook Integration
Kubernetes internal ports	2379/tcp 2380/tcp 6443/tcp 9099/tcp 10250/tcp 10251/tcp 10252/tcp	Required for normal kubernetes work

Managed Service Requirements

The Anodot customer must provide VPN SSH access to the cluster for metrics collection, remote monitoring and management, troubleshooting and support.
Anodot monitors the cluster remotely at all times by pushing KPIs (key performance indicators) to its data centers from the on-premise cluster. This makes sure your application is performing according to the KPIs that were initially defined. KPIs consist of metrics such as:

- Load averages and idle work
- IO rates
- Application error counts
- Latencies

Watching these metrics allows Anodot to identify issues as quickly as possible and alert customers if any problems are identified.



Hardware Requirements

Purchasing and maintaining hardware is the Anodot customer's responsibility. On-site installation is done by Anodot engineers.

Multi-Host or Single Host Requirements

Anodot clusters assume that all nodes are equal. Hosts are expected to be located on the same data center, with a latency of 1 millisecond between them. Larger latencies tend to worsen problems in distributed systems.

Hardware	Requirements
File System	2x Devices EXT4 Formatted Root File Server ● 90GB ● General Storage Data File Server ● 1TB ● SSD or Equivalent 5K IOs device
Network	1Gb Interface
Memory	At least 96GB
CPU	● 32 vCPU or 16 physical/dedicated threads ● High frequency CPU. For example, high frequency Intel Xeon E5-2666 v3 (Haswell) processor

Management Server Requirements

An additional management host machine must be available to store local Docker images.
This repository is used mainly for upgrades and updates. It is also used for the initial installation and cluster setup, and for recovery in case of cluster issues.

Management Server Specifications

- 8GB RAM
- 40GB Disk
- 2 CPU
- The same operating system and software as the Anodot nodes
- DNS A record/static IP [*Optional*] - This allows the server to be switched off when not in use

Agents Hardware Requirements

Additional hardware is required in order to install the agents collecting the data into Anodot: •
8vCPU 20Gb RAM - can handle 40 pipelines and 2000 eps

(See full details in the Installing Agents section)

Software Requirements

Initial setup and configuration are done by Anodot engineers. Installation of the software defined here, as well as system and data center monitoring, including host health, network health, and storage, is the Anodot customer's responsibility.

Supported Operating Systems

- Container Linux by CoreOS
- Debian Jessie, Stretch, Wheezy
- Ubuntu 16.04, 18.04
- CentOS/RHEL 7
- Fedora 28
- Fedora/CentOS Atomic
- openSUSE Leap 42.3/Tumbleweed

Specific Requirements per Operating System

System	Requirements
Ubuntu	● Python-apt ● Aufs-tools ● Apt-transport-https ● Software-properties-common ● Ebtables
Fedora	● Libselinux-python ● Device-mapper-libs ● Ebtables
Suse	● Device-mapper ● Ebtables
CentOS	● Libselinux-python ● Device-mapper-libs ● Ebtables ● NSS

Red Hat	● Libselinux-python ● Device-mapper-libs ● Ebtables ● NSS
Debian	● Python-apt ● Aufs-tools ● Apt-transport-https ● Software-properties-common ● Ebtables

Additional Dependencies

- Python 2 or Python 3
- Python-httplib2
- Curl
- Rsync
- Bash-completion
- Socat
- Unzip
- Docker-ce=18.09

Notice!

We recommend for Kubernetes / Docker installations to avoid the 15.0-45-generic version as it has a bug.

Load Balancer Setup

	Description
Protocol	HTTP
External Ports	443
Listener	Since Anodot nodes are symmetric, we recommend sending traffic to all nodes. Internal routing is performed to match the specific service-node.

DNS	Any FQDN
Certificate and SSL	We recommend TLS 1.2
	AES 256
Idle Timeout	300 seconds
Stickiness	NA
Health check	TCP (Kubernetes Ephemeral Port)

Backup and Restore

Backup and backup maintenance are the Anodot Customer's responsibility.

Anodot will maintain a daily data snapshot on a Persistent Volume provided by the Customer for up to 30 days. It is the Customer's responsibility to make sure that the volume is backed up and available in case a data restore is required.

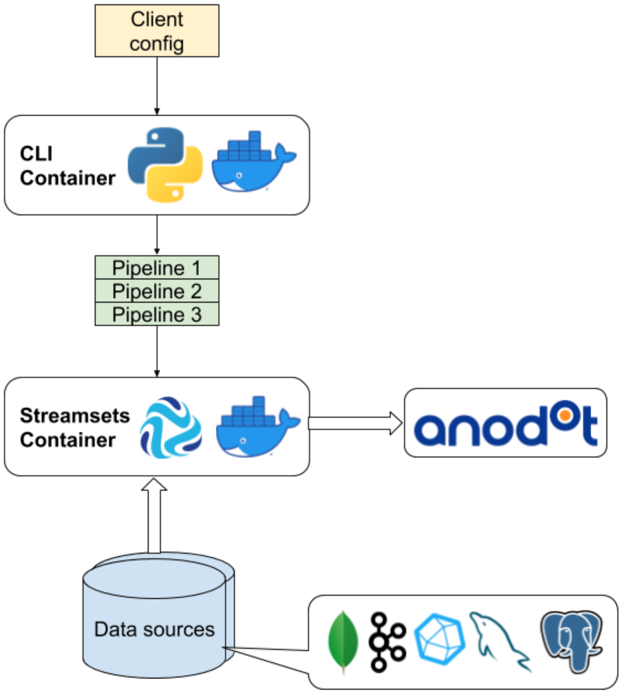
If a restore is required, data can be imported from the backup volume by Anodot engineers.

Getting Data using Anodot Agents

Anodot is using a set of software agents enabling it to consume metrics from the different on-premise data sources customers have. These agents are implemented as an Open Source library on github here - https://github.com/anodot/daria

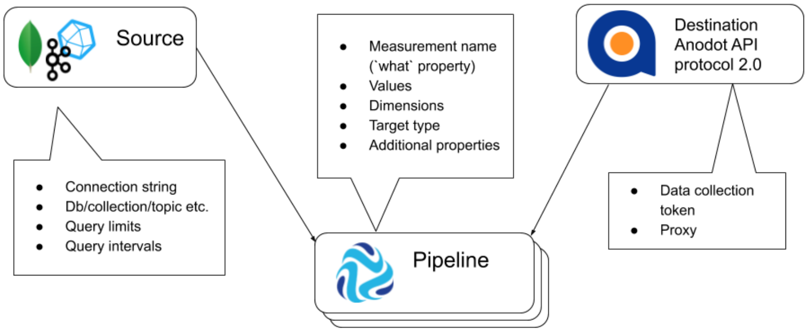
The documentation wiki there is always the most up to date and it is recommended to check periodically for updates there.

Main Concepts



Config Structure

- **Source** - This is where you want your data to be pulled from. Available sources: mongo, kafka, influx, mysql, postgres
- **Destination** - Where to put your data. Available destinations: *http client* - Anodot rest API endpoint
- **Pipeline** - pipelines connect sources and destinations with data processing and transformation stages



What pipelines do:

1. Take data from source
2. If destination is http client - every record is transformed to json object according to specs of anodot 2.0 metric protocol
3. Values are converted to floating point numbers
4. Timestamps are converted to UNIX timestamp in seconds

Basic flow

1. **Create a destination** (with Anodot API token, you can copy it from Settings > API tokens > Data Collection in your Anodot account). Also, here you can configure proxy for connecting to Anodot. You can also configure a proxy server for connecting to Anodot.

```
> agent destination
Anodot api token: tokenhere
Use proxy for connecting to Anodot? [y/N]: y
Proxy uri: http://squid:3181
Proxy username []:
Proxy password []:
Destination configured
```

Now you can check that Monitoring pipeline is running and some monitoring data is coming to Anodot

2. **Create source**

```
agent source create
```

3. **Create pipeline**

```
agent pipeline create
```

4. **Run pipeline**

```
agent pipeline start PIPELINE_ID
```

5. **Check pipeline status**

```
agent pipeline info PIPELINE_ID
```

6. If errors occur - check the troubleshooting section
   i. fix errors
   ii. Stop the pipeline agent pipeline stop PIPELINE_ID
   iii. Reset pipeline origin agent pipeline reset PIPELINE_ID
   iv. Run pipeline again

Accessing Agents

Once the agents are installed, you can access them using the following command in CLI:

```
kubectl exec -it streamsets-agent-0 -c agent bash
```

Where streamsets-agent-0 should be replaced with actual pod name

Breaking compatibility

If you upgrade from version <1.6.0 kafka pipelines will be deprecated. Whey will still be running but you won't be able to update them. You will need to delete pipelines, delete sources and recreate them with the new config according to the documentation.

Monitoring

When you create a destination, monitoring pipeline inside the agent is created. It pushes to Anodot some system metrics such as:

- cpu_load
- live_threads
- heap_memory_usage
- non_heap_memory_usage
- ConcurrentMarkSweep_collections
- ConcurrentMarkSweep_time
- ParNew_collections

- ParNew_time
- health_check - every 60 seconds monitoring pipeline must send health check with value 1 to indicate that it's working (normal value is 4-5 every 5 minutes)
- destination_latency - how long the request to Anodot API is executed

Metrics for each pipeline:

- pipeline_batchCount
- pipeline_batchErrorRecords
- pipeline_batchInputRecords
- pipeline_batchOutputRecords

Dimensions:

- source: agent_monitoring
- host_id - randomly generated string for each container
- pipeline_id - for pipeline metrics
- pipeline_type - for pipeline metrics

Tags:

- host_name - HOSTNAME environmental variable

You can also import a prebuilt monitoring dashboard

The total amount of metrics generated by monitoring: number of agents * 10 + number of pipelines * 5

Suggested alerts:

- Health check value is less than 3 (time scale 5 min)
- Memory usage and CPU usage based on your machine capacity

Troubleshooting

Pipelines may not work as expected for several reasons, for example, wrong configuration, or some issues connecting to the destination, etc. You can look for errors in three locations:

1. agent pipeline info PIPELINE_ID - This command will show some issues if a pipeline is misconfigured
2. agent pipeline logs -s ERROR PIPELINE_ID - shows error logs if any
3. Also sometimes records may not reach the destination because errors happened in one of data processing and transformation stages. To see these errors use this command:

   ```
   get_errors PIPELINE_ID
   ```

4. It's possible to enable logging of requests to Anodot and see the exact data being sent.

   i. Stop the pipeline agent pipeline stop PIPELINE_ID

   ii. Enable logging agent pipeline destination-logs --enable PIPELINE_ID

   iii. Start the pipeline agent pipeline start PIPELINE_ID iv. See logs destination_logs

   v. After troubleshooting stop the pipeline and disable logs because they consume a lot of space agent pipeline destination-logs --disable PIPELINE_ID