```
# Default configuration file for the Linux (deb/rpm) and Windows MSI
collector packages

# If the collector is installed without the Linux/Windows installer
script, the following
# environment variables are required to be manually defined or configured
below:
# - SPLUNK_ACCESS_TOKEN: The Splunk access token to authenticate requests
# - SPLUNK_API_URL: The Splunk API URL, e.g. https://api.us0.signalfx.com
# - SPLUNK_BUNDLE_DIR: The path to the Smart Agent bundle, e.g.
/usr/lib/splunk-otel-collector/agent-bundle
# - SPLUNK_COLLECTD_DIR: The path to the collectd config directory for
the Smart Agent, e.g. /usr/lib/splunk-otel-collector/agent-
bundle/run/collectd
# - SPLUNK_HEC_TOKEN: The Splunk HEC authentication token
# - SPLUNK_HEC_URL: The Splunk HEC endpoint URL, e.g.
https://ingest.us0.signalfx.com/v1/log
# - SPLUNK_INGEST_URL: The Splunk ingest URL, e.g.
https://ingest.us0.signalfx.com
# - SPLUNK_LISTEN_INTERFACE: The network interface the agent receivers
listen on.
# - SPLUNK_TRACE_URL: The Splunk trace endpoint URL, e.g.
https://ingest.us0.signalfx.com/v2/trace

extensions:
  health_check:
    endpoint: "${SPLUNK_LISTEN_INTERFACE}:13133"
  http_forwarder:
    ingress:
      endpoint: "${SPLUNK_LISTEN_INTERFACE}:6060"
    egress:
      endpoint: "${SPLUNK_API_URL}"
      # Use instead when sending to gateway
      #endpoint: "${SPLUNK_GATEWAY_URL}"
  smartagent:
    bundleDir: "${SPLUNK_BUNDLE_DIR}"
    collectd:
      configDir: "${SPLUNK_COLLECTD_DIR}"
  zpages:
    #endpoint: "${SPLUNK_LISTEN_INTERFACE}:55679"

receivers:
  fluentforward:
    endpoint: "${SPLUNK_LISTEN_INTERFACE}:8006"
  hostmetrics:
    collection_interval: 10s
    scrapers:
      cpu:
      disk:
      filesystem:
      memory:
      network:
      # System load average metrics
https://en.wikipedia.org/wiki/Load_(computing)
```

```
    load:
    # Paging/Swap space utilization and I/O metrics
    paging:
    # Aggregated system process count metrics
    processes:
    # System processes metrics, disabled by default
    # process:
jaeger:
  protocols:
    grpc:
      endpoint: "${SPLUNK_LISTEN_INTERFACE}:14250"
    thrift_binary:
      endpoint: "${SPLUNK_LISTEN_INTERFACE}:6832"
    thrift_compact:
      endpoint: "${SPLUNK_LISTEN_INTERFACE}:6831"
    thrift_http:
      endpoint: "${SPLUNK_LISTEN_INTERFACE}:14268"
otlp:
  protocols:
    grpc:
      endpoint: "${SPLUNK_LISTEN_INTERFACE}:4317"
    http:
      endpoint: "${SPLUNK_LISTEN_INTERFACE}:4318"
# This section is used to collect the OpenTelemetry Collector metrics
# Even if just a Splunk APM customer, these metrics are included
prometheus/internal:
  config:
    scrape_configs:
    - job_name: 'otel-collector'
      scrape_interval: 10s
      static_configs:
      - targets: ["${SPLUNK_LISTEN_INTERFACE}:8888"]
      metric_relabel_configs:
        - source_labels: [ __name__ ]
          regex: 'otelcol_rpc_.*'
          action: drop
        - source_labels: [ __name__ ]
          regex: 'otelcol_http_.*'
          action: drop
        - source_labels: [ __name__ ]
          regex: 'otelcol_processor_batch_.*'
          action: drop
smartagent/signalfx-forwarder:
  type: signalfx-forwarder
  listenAddress: "${SPLUNK_LISTEN_INTERFACE}:9080"
smartagent/processlist:
  type: processlist
smartagent/snmp:
  type: telegraf/snmp
  agents:  ["1x2.1x.xx.xx:162"]
  version: 2
  community: "public"
        fields:
            name: "panLcLogDurationSystem"
```

```
              oid: ".1.3.6.1.4.1.25461.2.1.2.6.1.2.3"
  signalfx:
    endpoint: "${SPLUNK_LISTEN_INTERFACE}:9943"
    # Whether to preserve incoming access token and use instead of
exporter token
    # default = false
    #access_token_passthrough: true
  zipkin:
    endpoint: "${SPLUNK_LISTEN_INTERFACE}:9411"

processors:
  batch:
  # Enabling the memory_limiter is strongly recommended for every
pipeline.
  # Configuration is based on the amount of memory allocated to the
collector.
  # For more information about memory limiter, see
  # https://github.com/open-telemetry/opentelemetry-
collector/blob/main/processor/memorylimiter/README.md
  memory_limiter:
    check_interval: 2s
    limit_mib: ${SPLUNK_MEMORY_LIMIT_MIB}

  # Detect if the collector is running on a cloud system, which is
important for creating unique cloud provider dimensions.
  # Detector order is important: the `system` detector goes last so it
can't preclude cloud detectors from setting host/os info.
  # Resource detection processor is configured to override all host and
cloud attributes because instrumentation
  # libraries can send wrong values from container environments.
  #
https://docs.splunk.com/Observability/gdi/opentelemetry/components/resour
cedetection-processor.html#ordering-considerations
  resourcedetection:
    detectors: [gcp, ecs, ec2, azure, system]
    override: true

  # Optional: The following processor can be used to add a default
"deployment.environment" attribute to the logs and
  # traces when it's not populated by instrumentation libraries.
  # If enabled, make sure to enable this processor in a pipeline.
  # For more information, see
https://docs.splunk.com/Observability/gdi/opentelemetry/components/resour
ce-processor.html
  #resource/add_environment:
    #attributes:
      #- action: insert
        #value: staging/production/...
        #key: deployment.environment

exporters:
  # Traces
  sapm:
    access_token: "${SPLUNK_ACCESS_TOKEN}"
    endpoint: "${SPLUNK_TRACE_URL}"
  # Metrics + Events
  signalfx:
    access_token: "${SPLUNK_ACCESS_TOKEN}"
    api_url: "${SPLUNK_API_URL}"
    ingest_url: "${SPLUNK_INGEST_URL}"
    # Use instead when sending to gateway
    #api_url: http://${SPLUNK_GATEWAY_URL}:6060
    #ingest_url: http://${SPLUNK_GATEWAY_URL}:9943
    sync_host_metadata: true
    correlation:
  # Logs
  splunk_hec:
    token: "${SPLUNK_HEC_TOKEN}"
    endpoint: "${SPLUNK_HEC_URL}"
    source: "otel"
    sourcetype: "otel"
    profiling_data_enabled: false
  # Profiling
  splunk_hec/profiling:
    token: "${SPLUNK_ACCESS_TOKEN}"
    endpoint: "${SPLUNK_INGEST_URL}/v1/log"
    log_data_enabled: false
  # Send to gateway
  otlp:
    endpoint: "${SPLUNK_GATEWAY_URL}:4317"
    tls:
      insecure: true
  # Debug
  debug:
    verbosity: detailed

service:
  telemetry:
    metrics:
      address: "${SPLUNK_LISTEN_INTERFACE}:8888"
  extensions: [health_check, http_forwarder, zpages, smartagent]
  pipelines:
    traces:
      receivers: [jaeger, otlp, smartagent/signalfx-forwarder, zipkin]
      processors:
      - memory_limiter
      - batch
      - resourcedetection
      #- resource/add_environment
      exporters: [sapm, signalfx]
      # Use instead when sending to gateway
      #exporters: [otlp, signalfx]
    metrics:
      receivers: [hostmetrics, otlp, signalfx, smartagent/signalfx-
forwarder, smartagent/snmp]
      processors: [memory_limiter, batch, resourcedetection]
      exporters: [signalfx]
      # Use instead when sending to gateway
      #exporters: [otlp]
    metrics/internal:
      receivers: [prometheus/internal]
      processors: [memory_limiter, batch, resourcedetection]
      # When sending to gateway, at least one metrics pipeline needs
      # to use signalfx exporter so host metadata gets emitted
      exporters: [signalfx]
    logs/signalfx:
      receivers: [signalfx, smartagent/processlist]
      processors: [memory_limiter, batch, resourcedetection]
      exporters: [signalfx]
    logs:
      receivers: [fluentforward, otlp]
      processors:
      - memory_limiter
      - batch
      - resourcedetection
      #- resource/add_environment
      exporters: [splunk_hec, splunk_hec/profiling]
      # Use instead when sending to gateway
      #exporters: [otlp]
```