# Tekton Pipeline Log Persistance

|  |  |
| --- | --- |
| **Fragestellung** | **Wie sollen Pipeline Logs vollständig und hochverfügbar gelesen und gespeichert werden?**  In Tekton sind die Logs ausschliesslich in den Pods der jeweiligen TaskRuns gespeichert. Nach Abschluss eines Pipeline Tasks sind diese nicht mehr vollständig abrufbar, da sie auf der Node nicht komplett vorgehalten werden. Um die Logs auch nach Löschung der Pods noch verfügbar zu haben, sammelt der ESTA Tekton Controller nach Beendigung eines Pipeline Runs die Logs aus allen Containern. Zu diesem Zeitpunkt sind diese aber schon nicht mehr vollständig. Die Logs müssen also bereits zur Laufzeit der Pipeline aus den jeweiligen Containers gezogen und persistent gespeichert werden.  Wie und mit welchen Hilfsmitteln kann dies nun am effizientesten und unterbrechungsfrei (HA) erreicht werden?  [ESTA-5317](https://flow.sbb.ch/browse/ESTA-5317) - ESTA Tekton Logging HA Konzept Closed |
| **Rahmenbedingung** | * Betrifft ausschliesslich Logs aus Tekton Pipelines resp. TaskRuns * Keine cluster-weite Logging Lösung gesucht * Am Cluster Setup soll wenn möglich nichts geändert werden müssen |
| **Annahmen** | * Logs eines PipelineRuns können mehrere 100MB gross werden |
| **Alternativen / Varianten** | Variante 1 **ESTA Tekton Controller streamed die Logs zur Laufzeit**  Der Controller hat bereits einen Watcher auf Pipeline Events und kann so beim Start einer Pipeline den Log Stream lesen und laufend ins lokale Dateisystem schreiben.   |  |  | | --- | --- | | **Vorteile** | **Nachteile** | | Einfache Umsetzung | Nicht HA | |  | Hoher Ressourcenverbrauch der Controllers |  Variante 2 **Ein einfacher Service streamed Logs Pod-basiert**  Einfach umzusetzende Lösung mit einem eigenen kleinen Service (Quarkus native) der einen Watcher auf PipelineRun Events registriert und bei jedem Start die Logs direkt von den Pods via fabric8 liest und pro Pod/Task in individuellen Dateien speichert. Die abgeschlossenen Logs werden gzipped und nach S3 hochgeladen.   |  |  | | --- | --- | | **Vorteile** | **Nachteile** | | Einfache Umsetzung | Nicht HA | | Deployment mit Tekton Helm Chart | Zusätzliches Deployment (Maintenance) |  Variante 3a **Logging nach Splunk (full)**  Bestehende Infrastruktur nutzen und die TaskRuns mit Annotations so konfigurieren, dass sie nach Splunk loggen. Die Logs müssen dann vom Tekton Control Panel resp. dem Backend aus Splunk gelesen werden. Optimalerweise wird für die Speicherung in Splunk ein spezifischer Index für die Applikation (Tekton Namespace) verwendet, welcher explizit bestellt und konfiguriert wird. Siehe dazu  [ESTA-5185](https://flow.sbb.ch/browse/ESTA-5185) - Esta Tekton Optional Splunk Logging Open   |  |  | | --- | --- | | **Vorteile** | **Nachteile** | | HA | Teuer | | Einfach einzurichten | Refactoring des Controllers oder UI | | Kosten beim Kunden | Lifetime nicht in Sync mit Tekton Config | |  | Access von Tekton auf Splunk Logs? |  Variante 3b **Logging nach Splunk (light)**  Basierend auf ESTA-5185 kann vom Kunden ein Splunk Index konfiguriert werden, in welchen die Pipeline Logs geschrieben werden. Ist ein Splunk Index konfiguriert, wird im Tekton Control Panel beim PipelineRun ein Link auf ein Search Query in Splunk angezeigt, ähnlich wie diese in der Openshift Console gemacht wird.  Das bisherige Logging in Tekton wird beibehalten mit dem Verweis, dass die Openshift Infrastruktur das Logging nur im aktuellen Rahmen ermöglicht. Wer grosse und vollständige Logs will, muss Splunk aktivieren (opt-in).   |  |  | | --- | --- | | **Vorteile** | **Nachteile** | | HA | Default Logging bleibt unvollständig | | Einfach einzurichten | Keine perfekte Integration in Tekton Control Panel | | Kein Refactoring nötig |  | | Kosten beim Kunden |  |  Variante 4 **Log Deamon mit fluentd und S3 als Output**  Quellen:   * [Fluentd <> K8S integration via annotations](https://medium.com/@nachomillangarcia/fluentd-k8s-integration-via-annotations-9a035129f463) * [Application Logging in Kubernetes with fluentd](https://medium.com/@joatmon08/application-logging-in-kubernetes-with-fluentd-4556f1573672) * [Cluster-level Logging in Kubernetes with Fluentd](https://medium.com/kubernetes-tutorials/cluster-level-logging-in-kubernetes-with-fluentd-e59aa2b6093a)  |  |  | | --- | --- | | **Vorteile** | **Nachteile** | | HA | Refactoring des Controllers | | Native Lösung | Benötigt Node-Konfiguration (Deamonset) |  ~~Variante 5~~ **Openshift Cluster Logging mit Elasticsearch**  Quellen:   * [Logging subsystem for Red Hat OpenShift](https://docs.openshift.com/container-platform/4.11/logging/cluster-logging.html) * [Elastic Observability mit Elastic Cloud on Kubernetes (ECK)](https://discuss.elastic.co/t/dec-16th-2020-en-monitoring-tekton-tasks-and-pipelines-with-elastic-observability/257567)  |  |  | | --- | --- | | **Vorteile** | **Nachteile** | | Supportete Lösung von RedHat | Ganzer Elasticsearch Stack auf Openshift | |  | Wird von CON nicht approved | |
| **Entscheidung** | **Variante 1** (mit evtl. Migration nach Variante 2) |
| **Begründung** | Einfachste und schnellste Umsetzung zum aktuellen Zeitpunkt. Kann schrittweise mit Storage auf S3 (→ [ESTA-5267](https://flow.sbb.ch/browse/ESTA-5267)) und Leader-Election zu einer HA-Lösung erweitert oder bei Bedarf in einen eigenen Service (→ Variante 2) ausgelagert werden. |
| **Wer** | [Spirig Lukas (IT-PTR-CEN2-SL3)](file:///C:\display\~u228304) , [Wallrapp Manuel (IT-PTR-EXT-EXT2 - Extern)](file:///C:\display\~ue64007) , [Jeanneret Julien (IT-PTR-CEN1-BDE3)](file:///C:\display\~u231154) , [Brüderli Thomas (IT-PTR-CEN2-SL2)](file:///C:\display\~u233658) , [Härtwig Mario (IT-PTR-CEN2-BDE5)](file:///C:\display\~u237004) , [Burger Alain (IT-PTR-CEN2-YPT4)](file:///C:\display\~u236168) , [Wattinger Matthias (IT-PTR-CEN1-YPT1)](file:///C:\display\~u239086) |
| **Wann** | 24.11.2022 |