stonebranch

Red Hat
Certified Technology

Application Automation For OpenShift

Seamlessly Orchestrate Data Supply Chains

Table of Contents

1. Introduction

Stonebranch builds IT orchestration and automation solutions that transform business IT environments from simple IT task automation into sophisticated, real-time business service automation. No matter the degree of automation, the Stonebranch platform is simple, modern, and secure. Using the Stonebranch Universal Automation Platform, enterprises can seamlessly orchestrate workloads and data across technology ecosystems and silos. Headquartered in Atlanta, Georgia, with points of contact and support throughout the Americas, Europe, and Asia, Stonebranch serves some of the world's largest financial, manufacturing, healthcare, travel, transportation, energy, and technology institutions.

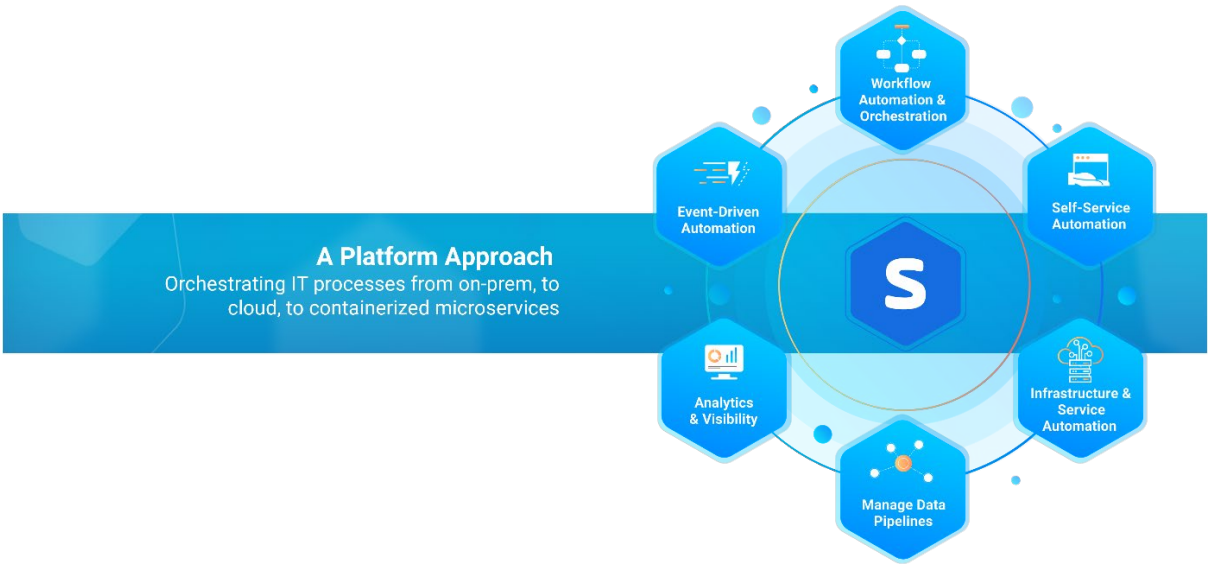

Figure 1: Real-Time Hybrid IT Automation Platform - Overview

Stonebranch offers the most modern real-time IT automation and orchestration platform designed to centrally manage and orchestrate automated jobs, tasks and workflows across hybrid IT environments from on-prem to the cloud. Within the UAC platform, we have five key solution areas. The focus is to provide an expandable and future proof solution where you can automate and orchestrate anything you need from a single platform.

Figure 2: Key Solution Portfolio

2. Scope of the Document

The intention of this whitepaper is to provide a solution for the following questions:

- How to securely transfer business data located on the mainframe, in cloud storage, in an SAP system, or any (virtual) server, to an application running on OpenShift (and vice versa)
- How to automatically trigger a file transfer in real-time based on various events like a file arrival, an email, a message in the message queue, or based on time
- How to trigger the file transfer from any application calling a REST web service
- How to successfully implement cluster management - secure distribution of data to all PODs in an OpenShift cluster
- The need for real-time monitoring and auditing of the entire file transfer process
- How to functionally integrate applications running in OpenShift into the legacy IT landscape

In this whitepaper we explain how to securely transfer business data located on the in the cloud, on the mainframe, or in a hybrid or homegrown environment to an application running on OpenShift, and vice versa, on demand. The resulting benefit will be that the applications on OpenShift will always have the latest and most up-to-date business data. The solution also supports cluster scenarios, meaning that you can provide the data simultaneously to all PODs related to an application, or only to the one with the lightest current load (round robin or other scenarios are also supported).

In addition to the data transfer scenario, the solution also allows users to schedule their applications running on OpenShift in the same way as applications running in the cloud, on a (virtual) server, or on the mainframe ,including all the benefits of an application deployed on the OpenShift orchestration platform like highly reduced resources consumption, scalability and performance, fast deployment and testing. You can additionally integrate your applications on OpenShift into any current business process automation flow consisting of both OpenShift and non-OpenShift applications, such as an SAP order-to-cash process.

The solution will be explained using a practical example from the AXA insurance.

3. Use Case

In order to increase sales and new customers acquisitions, modern companies must be able to, quickly launch new applications in their self-service portal through digital, easy access channels as part of their digital transformation strategy. Such an approach would additionally decrease costs through automation and an enhanced, self-propelled customer experience. Essentially any existing business processes that are inefficient should therefore be replaced with automated digital solutions, and new business processes should be designed to be web self-service enabled from the beginning. They should also be scalable, always available, and cost-efficient.

When leveraging such a self-service portal as a more efficient way of doing business, it is key that the applications provided in the portal provide a consistent quality of service and have the required business data available in-time. The goal is to provide a consistent user experience, regardless of when the customer is using the application or how many customers are concurrently using it.

3.1. Consistent Quality of Service

To provide a consistent and cost-efficient quality of service, it is important that the offered applications can scale in horizontal elastic way. This means that the computing resources provided should be capable of scaling up to accommodate an increasing or decreasing number of customer requests.

Such an "elastic" scalability functionality can be provided with relatively little disruption by deploying the portal applications on an OpenShift container platform. The OpenShift platform allows you to build, deploy and manage your container-based applications consistently across cloud and on-premises infrastructure. The applications are deployed in OpenShift PODs*: "A POD is one or more container deployed together on one host, and the smallest compute unit that can be defined, deployed, and managed." [1]*

The OpenShift orchestration platform allows to dynamically scale up and down the number of application PODs to provide a consistent resource with efficient quality of service for the application user.

3.2. Just-In-Time Availability of Business Data

It is also key that the business data required by the application and offered through the portal is consistent and up-to-date. If three instances of an application are started in parallel, then all three instances should receive the same business data at the same time.

Often, business data needs to be collected from not one, but various sources. Requested data may need to be aggregated from, for example, a combination of an SAP system, a database, a Linux/Windows server, cloud storage, and even a mainframe; it must then be transferred to the applications running in an OpenShift POD. In many cases, it is also a requirement to provide this aggregate data to all relevant applications as soon as it is available, rather than waiting until a specific time of the day is reached.

Stonebranch Universal Automation Center provides such an event /real-time based data supply chain process, offering out of the box dedicated support for transferring data from any source to one or multiple OpenShift POD, in parallel. as out of the box functionality.

3.3. Universal Automation Center

Universal Automation Center is a web-based enterprise scheduler which is available for on-premise functionality or in the AWS cloud. Universal Automation Center is fully event-based. It consists of the Universal Controller, a web-based workflow, reporting and orchestration engine, and a workload execution component known as Universal Agent. As soon as an agent is installed on a server, it automatically connects to the Universal Controller

middleware message bus OMS and is ready for executing commands/scripts and file transfers. For applications that provide an API, no agent needs to be installed.

The Universal Agent runs on any OS platform, including IBM mainframe. For OpenShift, a dedicated OpenShift image containing the Stonebranch Universal Agent is provided through the Docker hub registry. The OpenShift image is deployed as a sidecar container to any POD running an application. As soon as a POD is started, the sidecar container is automatically initiated and the contained Universal Agent automatically connects to Universal Automation Center message bus (OMS). During the start of the POD, Universal Agent is automatically assigned to a Universal Controller application cluster, providing various cluster functionalities like load-balancing or broadcast cluster, which allows users to send a file or command to all agents in the cluster simultaneously. One dedicated Universal Controller application cluster is created for each OpenShift application. As soon as the Universal Agent registers at the Universal Controller message bus, it is available to send and receive files from any other Universal Agent installed on a server or mainframe within the IT landscape, or agent-less from any cloud storage. In addition, the application running in the POD can be scheduled like any other application and included in any business process automation workflow.

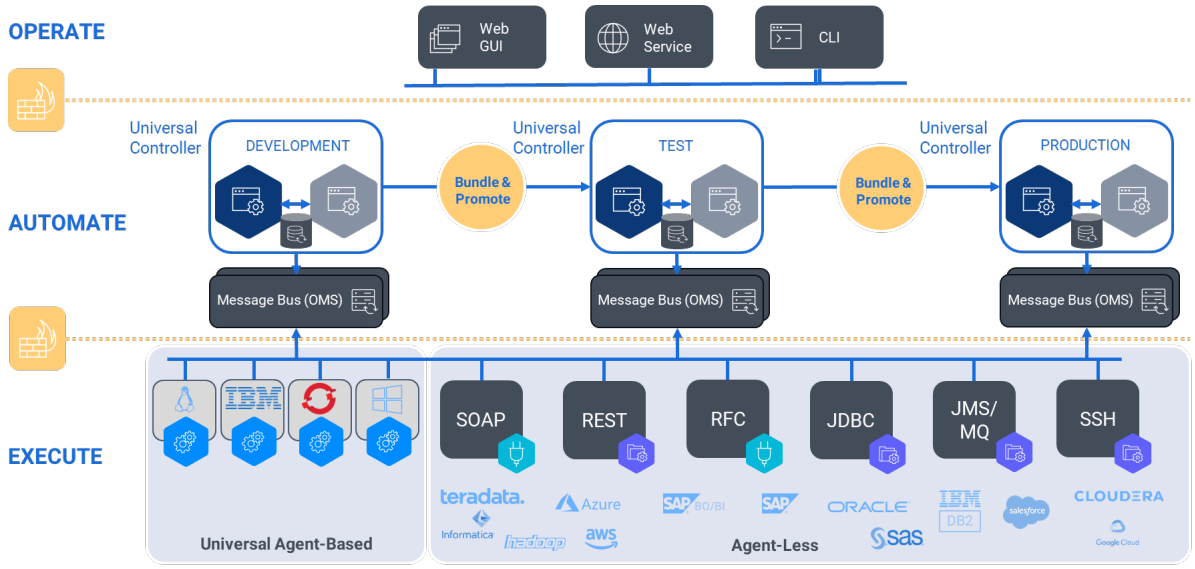

Figure 3: Universal Automation Center, Architecture

DESCRIPTION:

From an architectural point of view, the UAC components fulfill the following roles:

Component
Universal Controller (UC)
• Web application with user and administration GUI • Business logic to control workflows and processes • Persist UAC system state • Exposes RESTful APIs and CLI for headless operation • agentless integration to orchestrate web services and DBs
Message Bus (OMS)
• Messaging middleware • Inbound connections from clients
Universal Agent (UA)-Based Scheduling
• Executes and monitors workload • Performs managed file transfer (MFT) • Connects to Universal Controller via OMS • Exposes CLI to execute workload from any scheduler • Installed on wide variety of operating systems (MF, server, virtual server, container) • Tight integration with SAP for full application automation
Agentless Scheduling
• Applications providing an API are scheduled agent-less using the provided protocol (SOAP, REST...) • Opensource marketplace with many pre-integrated application connectors is available in GitHub • Dedicated certified connector is provided For SAP and PeopleSoft
Bundle & Promote – Lifecycle Management
• Bundle and Promote allows for the automatic moving of tested configurations from DEV -> TEST -> PROD – Supporting any C:D DevOps scenarios.

Table 1: Description Components

3.4. Use Case Description

The following is a real customer use case from the AXA insurance.

A world-leading insurance company provides its agents and customers with various insurance-related self-service applications on their web portal. To ensure that applications run stably even with a larger number of parallel requests, they are set up in a highly scalable OpenShift environment hosted on-premise and on public clouds.

The various applications providing different insurance functions are run in containers in an OpenShift POD. These insurance applications require business data from multiple sources, including the mainframe and various public cloud storage systems. The applications not only

receive data, but also provide data to connected systems like SAP business warehouse to push contract data through.

3.5. Requirements

The following are the requirements that the solution had to support:

- Transfer a file from the mainframe to all started OpenShift PODs related to an application
- Transfer a file from an application in OpenShift to an application within the internal IT landscape, e.g. SAP business warehouse
- Transfer a file from any cloud storage to all OpenShift PODs related to an app (and vice versa)
- Trigger a file transfer from a 3rd party web app located in an OpenPaaS landscape
- Solution that is cloud-based, secure and high available

3.6. Solution Architecture

Building off, of the above requirements for the solution, the following architecture was implemented. The figure below outlines how the data is transferred from the mainframe or cloud storage to all started instances of an application. Each instance of an application is represented by one POD.

Figure 4 below shows two applications: MyAXA-Business and MyAXA-WEB. Each application is represented by one POD. Depending on the web portal load, the number of application instances is scaled up or down by starting or stopping additional PODs per application. In our example two application instances (2 PODs) have been started already.

All PODs contain a sidecar container with a Stonebranch Universal Agent. Once a POD is started, the sidecar container is also started, and the Universal Agent of the sidecar container automatically registers to a Universal Controller agent cluster specific to that application. One Universal Controller agent cluster is created for each application, containing the agents of all started instances of the application. In the example below, two Universal Controller agent clusters have been started: one cluster for application one containing two active agents (= two started PODs) and one cluster for application two also containing two active agents (= two started PODs).

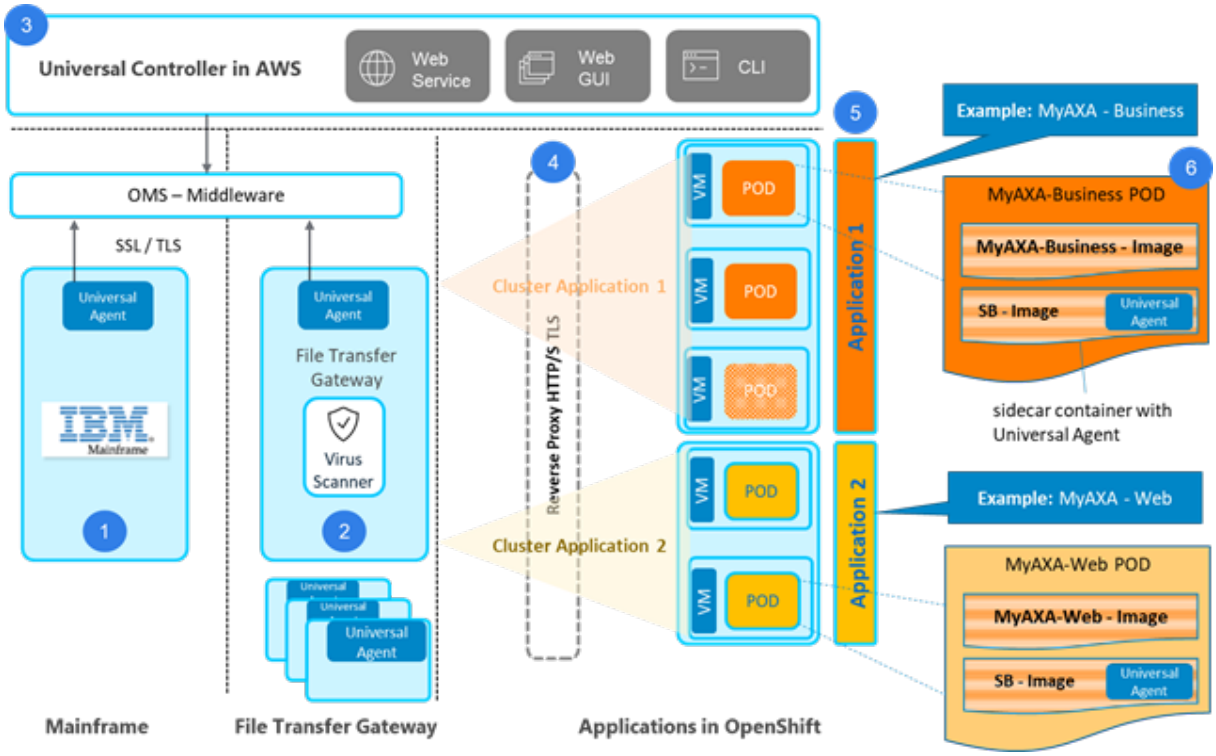

Figure 4: Provide Business Data from the mainframe to an application running in a POD

DESCRIPTION:

N#	Component	Description
Customer Data Centre		
1	IBM mainframe	IBM mainframe with Universal Agent installed
2	File Transfer Gateway Server (Linux)	File Transfer Gateway Server (Red Hat Linux) with Universal Agent installed. Each file transferred from the mainframe to the OpenShift PODs is first transferred for virus scan to the File Transfer Gateway (FTG). From FTG, the file is sent after successful Virus scan to one or all PODs assigned to the Universal Controller agent cluster. (**Note**: The intermediate transfer to the FTG is due to security regulations and not a technical limitation of the solution. You could also directly transfer from the MF to OpenShift or any other Node)
AWS Cloud		
3	Universal Automation Center with Universal Controller workflow engine	Universal Automation Center is a Web-Based Enterprise Scheduler, which is also available in the AWS Cloud. Universal Automation Center is fully event-based. It consists of the Universal Controller, a web-based workflow, reporting and orchestration engine and the Universal Agent as workload execution component. As soon as an agent is installed on a server, it automatically connects to the Universal Controller middleware message bus OMS and is ready for executing commands/scripts and file transfers. For applications that provide an API, no agent is needed.

N#	Component	Description
OpenShift PaaS		
4	Reverse Proxy	Due to security regulations, all communication from and to OpenShift should go via a reverse https proxy
5	Application Instance cluster	An application is deployed in a POD. When load increases, multiple instances can be started from the application. This means that multiple PODs are therefore started. When load decreases, application instances are stopped, meaning PODs are then stopped in OpenShift.
6	POD with Sidecar Container	All PODs contain a sidecar container with a Stonebranch Universal Agent. The side-car container is based on the Red Hat UBI image with a Universal Agent installed in it. The latest Version of the Image can be retrieved from the docker registry. Once a POD is started, the sidecar container is also started and the Universal Agent of the container automatically registers to a Universal Controller agent cluster dedicated for the application. For each application, one agent cluster is created, containing the agents of all started instances of the application.

Table 2: Description Solution Architecture

UNIVERSAL CONTROLLER AGENT CLUSTER

In Figure 5 are the agent clusters in Universal Controller. The agent cluster of application one contains two active agents (agent: image-ua-1-8vk9f - AGNT0044 and agent: image-ua-1-jq8d9 - AGNT0045).

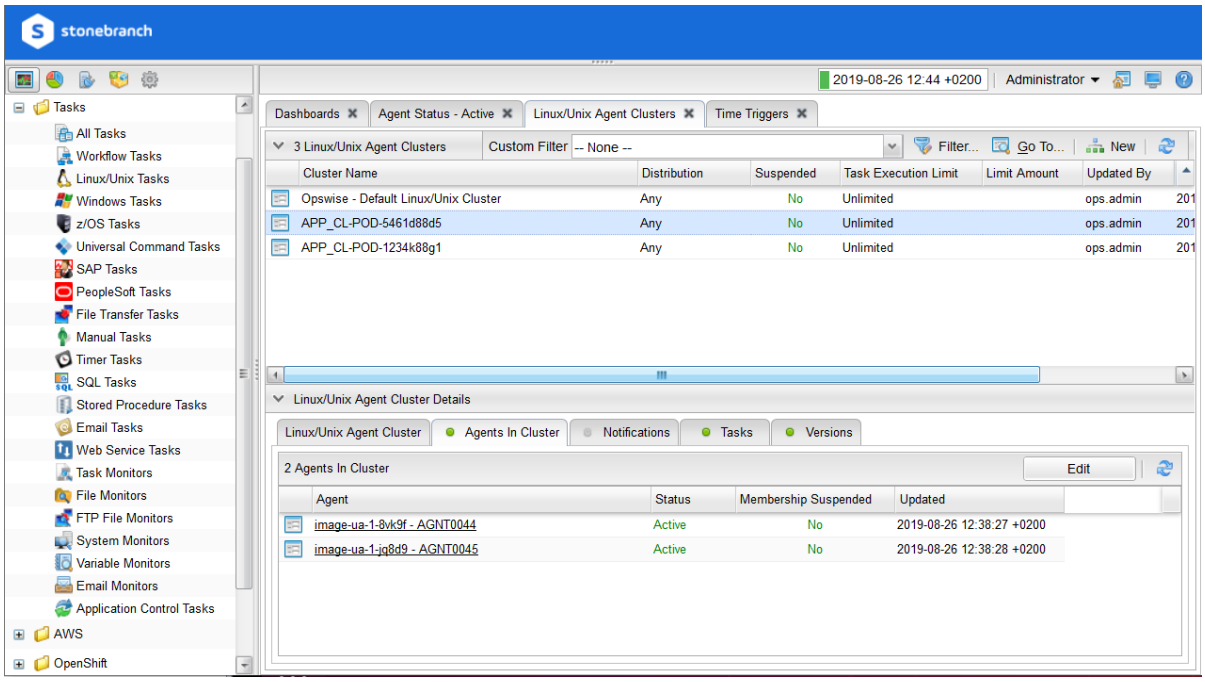

Figure 5: Universal Controller Agent cluster – Containing all Started PODs "Application Instances"

As soon as the Universal Agent of the sidecar container is assigned to the related Universal Controller agent cluster, all related PODs can send and receive files from any other Universal Agent installed on a server or mainframe within the insurance (several thousand Universal Agents are installed at the insurance company) IT landscape, or agent-less from any cloud storage. In addition, the application running in the POD can be scheduled like any other application and included in any business process that needs automating. The Universal Agent cluster supports sending files to just one agent (POD), or to all agents (meaning started PODs related to an application) in the agent cluster. depending on the business need.

If the web-portal load increases, the OpenShift orchestration platform allows users to dynamically scale up the number of application instances by starting an additional POD. When doing this, all new started PODs are automatically added to the Universal Controller agent cluster related to the application. Figure 6 shows this scenario in action. The number of PODs is increased in OpenShift from two to four. At the same time, you see in Figure 6 that the Universal Agents installed in the PODs are added dynamically to Universal Controller agent cluster.

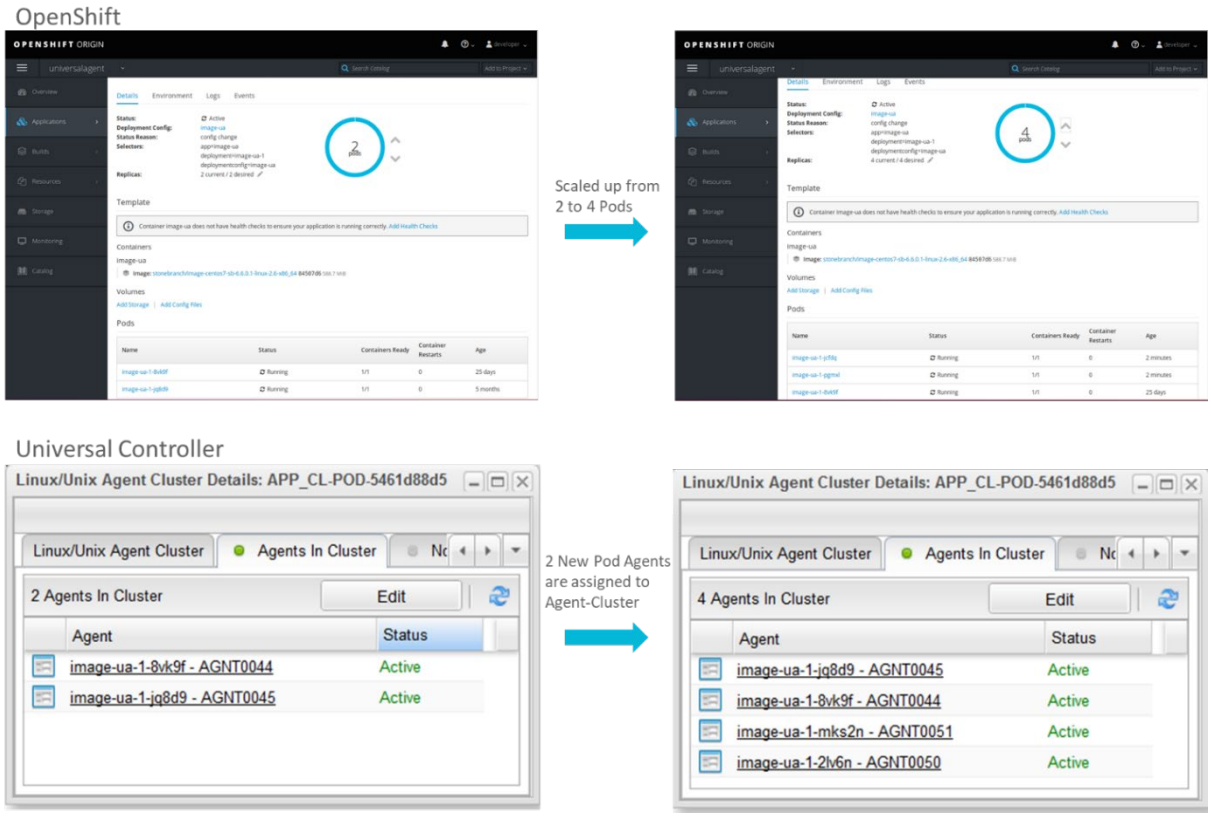

Figure 6: Scaling Up from 2 to 4 Application Instances in OpenShift and Automatically in Universal Controller

3.7. File Transfer Scenario

Below is an explanation of how a file is transferred from the mainframe to one or multiple application instances started for a specific application (POD).

First, it is important to note that a file transfer scenario to OpenShift is configured in the same way as a transfer to any other server in the Universal Controller web GUI.

In this particular insurance use case, it was required to provide data from the mainframe to all application instances belonging to the same application. Figure 7 shows the workflow configuration in the Universal Controller web GUI. A workflow is configured consisting of 5 Tasks. The first task, “openshift_mf_2_server,” then transfers the data from the mainframe to an intermediate file transfer gateway (refer also to Figure 2). On the file transfer gateway server, (FTG) the enterprise virus scanner is started for the received data by the task, “VirusScan”. If the virus scan was successful, the received data is moved to the folder “clean” using the task, “move_to_clean_for_OpenShift_upload”. If the move was successful, the task, “UDM_Linux_to_OpenShift Cluster_${ops_agent_name}” is started and loads the data to all PODs started for a specific application. All assigned PODs of an agent cluster (see Figure 5), therefore, will get the same data at the same time.

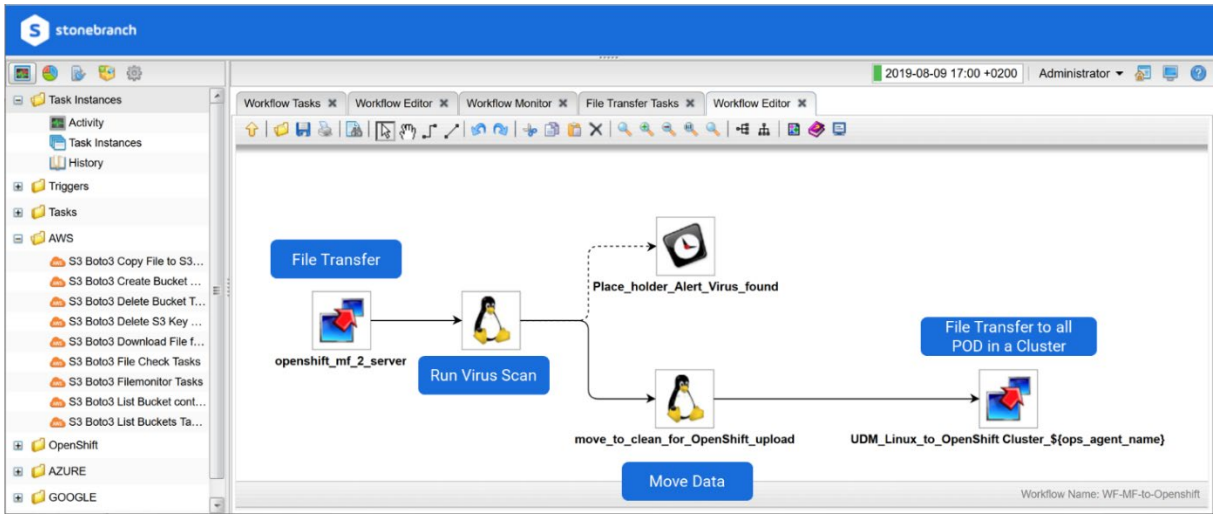

Figure 7: Workflow “Transferring a File from the Mainframe to OpenShift”

Figure 8 below shows the executed workflow in Universal Controller. For each Task, the execution status is visible in a real-time view.

The File Transfer Task “UDM_Linux_to_OpenShift Cluster_${ops_agent_name}” has been automatically spawned to one task instance per agent in the cluster. Two PODs had been started for the application; therefore, two file transfer instances are shown:

UDM_Linux_to_OpenShift Cluster_image-ua-1-jq8d9 - AGNT0036

UDM_Linux_to_OpenShift Cluster_image-ua-1-8vk9f - AGNT0037

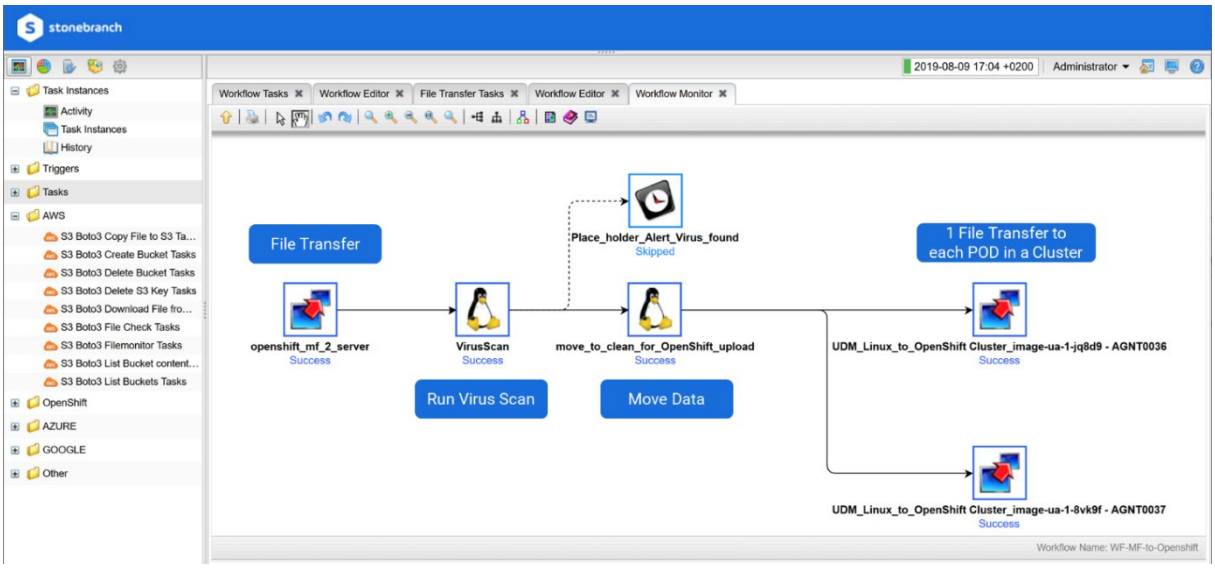

Figure 8: Active Workflow Instance "Transferring a File from the Mainframe to OpenShift"

3.8. Options to Trigger the File Transfer Scenario

It is vital that the business data required by the application and offered through the portal is consistent and up-to-date. If three instances of an application are started in parallel, then all three instances should get the same business data at the same time as soon as it is available on the mainframe.

Universal Controller supports triggering file transfers in various ways, including:

- Based on a file arrival
- Time-based (with support for internal and external calendar like an SAP calendar)
- Based on email arrival
- Based on a web service
- Event in a message queue (e.g. MQ, JMS)
- Based on the status of another task/workflow e.g. start a new file transfer, if the transfer from last night was successful

In this particular use case, the data is transferred in real-time based on a file arrival on the mainframe, or time-based, depending on the business requirement.

3.9. Integration of File Transfer into Microservices Architectures

Any file transfer can be triggered by calling the REST API of the Universal Controller. Essentially, you can start a file transfer workflow from any application, independent of whether it runs on a (virtual) server, mainframe or in a POD. This single API will allow you to perform a loose coupled integration into your microservices architecture.

3.10. Security

The file transfer is based on Stonebranch UDM protocol, which encrypts all data and communication channels using TLS1.2 (e.g. AES 256 / SHA 384).

4. How to Try it Out

To securely try out the transfer of business data located on the mainframe, in the cloud, in an HDFS cluster or any equal (virtual) server from or to an application running on OpenShift, only two steps are required:

1. Deploy the Red Hat certified Universal Agent container image (provided through the public Docker hub registry) as sidecar container to any POD to which you want the data transferred.
2. Request a Universal Controller cloud account on stonebranch.com.

If you want to transfer data from cloud storage, HDFS cluster to a POD, no additional Universal Agent is required. If you want to transfer data from a mainframe or (virtual) server, you will need to install a Universal Agent on each (virtual) server or mainframe. The Universal Agent can be downloaded from stonebranch.com.

5. Solution Summary

The following is a summary of the key solution components:

FUNCTIONALITIES PROVIDED:

- Secure transfer of business data located on the mainframe, cloud storage, SAP system or any (virtual)server to an application running on OpenShift, and vice versa
    - Automatic real-time triggering of file transfers based on a file arrival or time-based
    - File transfers triggered from any application calling a REST web-service
    - Cluster management - Secure distribution of data to all started PODs assigned to a cluster
    - Real-time monitoring and auditing of the entire file transfer process
    - Central logging of all activities – audit-proof
    - Lifecycle management - bundle and promote automatically moves tested configurations from DEV -> TEST -> PROD – supporting a C:D DevOps approach
- Cloud-ready, “web-based” solution – Universal Controller is available as SaaS in AWS
- High available solution using an AWS database with 3 availability zones
- In case of a crisis, a switch to another geographic region is supported
- All communication data and communication channels are TLS1.2 secured
- SAML Authentication for all Web GUI users
- All PODs contain a sidecar container based on a Red Hat UBI image with a Universal Agent

- The latest image is available via private docker store to ensure continuous updates of the Universal Agent and related Red Hat UBI image.
- Once a POD is started, the sidecar container is also started, and the Universal Agent of the sidecar container automatically registers to a Universal Controller Agent Cluster specific for the application. For each OpenShift application, one Universal Controller Agent Cluster is created.
- As soon as the Universal Agent of the sidecar container is assigned to the related Universal Controller Agent Cluster, all related PODs are able to send and receive files from any other Universal Agent installed on a server, either on the mainframe or agent-less from any cloud storage.
- Ability to schedule the application running in the POD from Universal Controller like any other application and included in a scheduling workflow

6. References

Ref-Id	Title	Filename	Remark
[1]	POD Definition	https://docs.openshift.com/enterprise/3.0/architecture/core_concepts/pods_and_services.html#pods	Red Hat documentation link

About Stonebranch

Stonebranch builds dynamic IT automation solutions that transform business IT environments from simple IT task automation into sophisticated, real-time business service automation, helping organizations achieve the highest possible Return on Automation.

No matter the degree of automation, Stonebranch software is simple, modern and secure. Using its universal automation platform, enterprises can seamlessly orchestrate workloads and data across technology stacks and ecosystems.

Headquartered in Atlanta, Georgia with points of contact and support throughout the Americas, Europe, and Asia, Stonebranch serves some of the world's largest financial, manufacturing, healthcare, travel, transportation, energy, and technology institutions.

www.stonebranch.com

Read what our customers are saying

IT Central Station
Unbiased reviews from the tech community