vFunction Operator Guide

Introduction

vFunction is a cloud-native modernization platform that combines dynamic and static code analysis, machine learning, and automation, to automatically identify and extract services from existing applications. vFunction is the only platform purpose-built for the modernization of Java applications.

The vFunction modernization process starts by learning the running monolithic application, and surfacing the interdependencies within it. Using AI, the platform analyzes and identifies services that can be separated from the application. This decomposition can present a range of micro, mini, or even macro services, depending on your application environment, each being an independently deployable and scalable application component.

vFunction automates the extraction of these services, enabling you to modernize your monolith, quickly and easily.

The vFunction platform

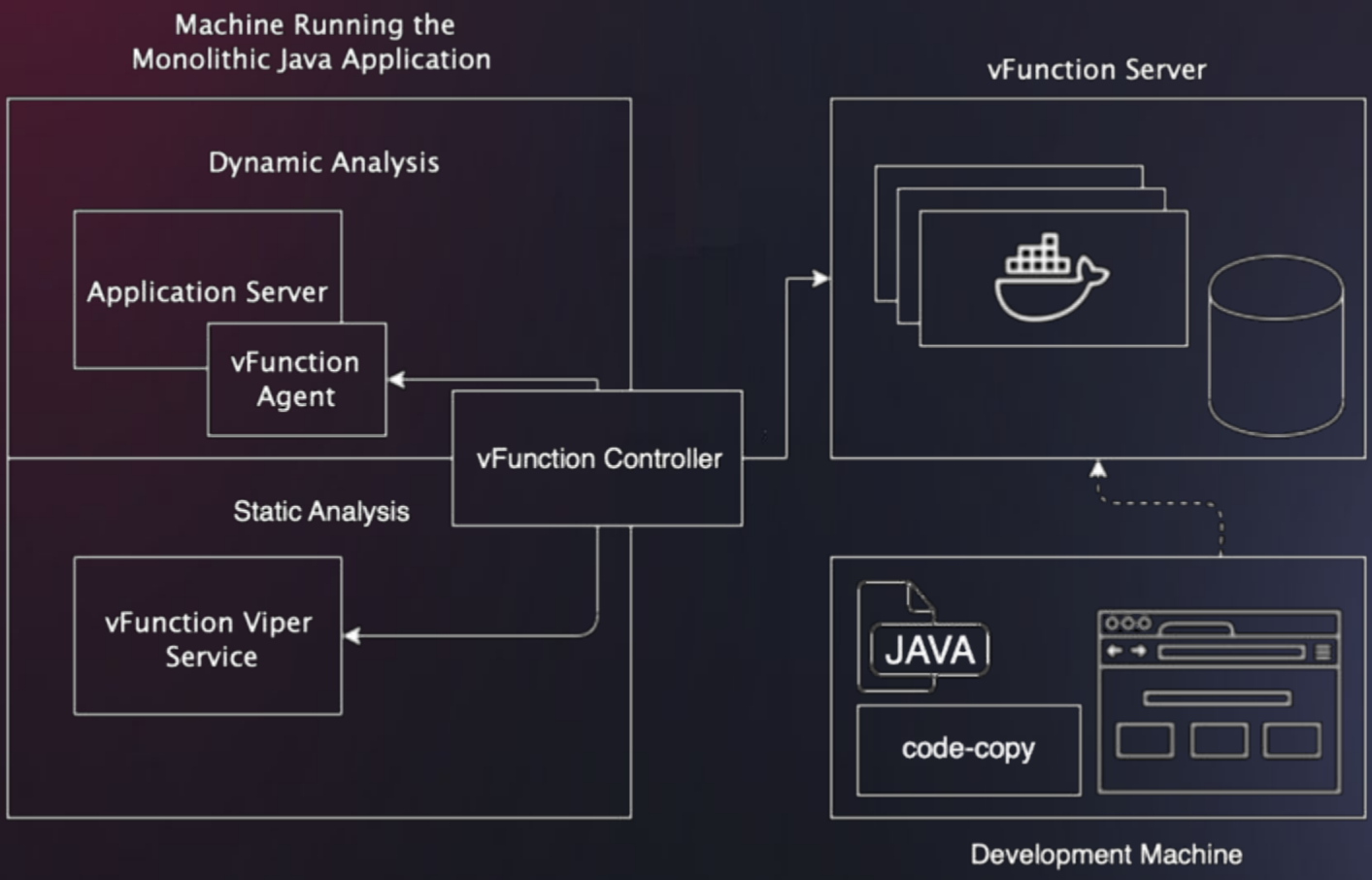
Review the diagram on the following page to familiarize yourself with the various components of the vFunction platform. The platform consists of 3 basic components; the server, the controller package, and a tools package. The server runs as an operand on an OpenShift environment. The controller package is installed on the machine that runs the monolithic application which can be either a Linux or a Windows machine, and the tools are run on a development machine, with access to the code of the monolithic application.

The controller package consists of three elements: the vFunction agent, that collects data during the dynamic analysis phase; the vFunction Viper application, that performs static analysis on the binaries of the application; and the vFunction controller that handles all the communication between the agent, Viper, and the vFunction server.

The vFunction agent is a mix of a Java and native agent, and needs to run on the JVM that is currently running your application. Refer to the vFunction OOB Support Matrix document for a list of supported application servers and JVMs.

vFunction

The vFunction components



Before you begin

The vFunction operator requires Red Hat OpenShift Kubernetes Application Platform 4.x.

The recommended cluster configuration (also the minimal) should consist of:

- 1 master node
- 1 worker node
- Storage capacity - enough to dynamically provision 2 PVs of 50G each
- A configured default StorageClass

Encryption: vFunction doesn't provide any encryption mechanism for data stored on any attached storage. If required, encrypt your data using an external encryption mechanism within the attached storage.

Backup: vFunction supports the backup of all critical data as part of the application. That said, we recommended that you apply your own storage-based backup mechanism to ensure complete backup of all data.

Health & Scalability: As a non-critical analysis software solution, the system is not required to scale. When recovery/disruption occurs, the vFunction application relies on kubernetes to scale or re-deploy the workload based on replica sets. The application doesn't monitor application health but rather relies on kubernetes to reschedule dead nodes and pods. Since only a single node is required for the replica set, the system gracefully recovers as the nodes are stateless.

Installation prerequisites

Cluster:

The operator was certified on OCP 4.5 and tested on OCP 4.3 running on AWS cloud.
If you encounter any issue with other OCP versions or cloud providers, contact info@vfunction.com.

Storage:

Make sure that the system has a default StorageClass in place. During installation, the operator creates two new PersistentVolumeClaims (PVCs), which requires the default StorageClass to **dynamically provision** two PersistentVolumes (PVs).

Both PersistentVolumes are accessed with ReadWriteOnce mode.

Project:

The operator should be installed in a new and dedicated project (namespace). If you intend to install more than one vFunction operands in the same cluster, use a new project for each one. Installation of more than one operand on the same cluster allows scaling by load balancing multiple applications to different vFunction servers.

Install the vFunction operator

Our operator installs an instance of the vFunction server that you can connect to with 1 or more of the vFunction controllers installed on your application machine(s).

1. Prepare the yaml file to use in the installation:
   a. Copy the following yaml template into a text editor:

```
apiVersion: vfunction.com/v1
kind: VfunctionServer
metadata:
  name: vfunction
  namespace: vfunction
spec:
  host: "http://my.domain.com"
  org_name: "MyCompany"
  admin:
    email: "admin@mycompany.com"
    name: "Admin"
    password: "Password1!"
  upgrade: "Daily"
  smtp:
    password: ""
    url: ""
    identity: ""
    user: ""
  tls:
    crt: |
      -----BEGIN CERTIFICATE-----
      ...
      -----END CERTIFICATE-----
    key: |
      -----BEGIN PRIVATE KEY-----
      ...
      -----END PRIVATE KEY-----
```

   b. Replace the customizable template fields with your information:
      - **host**: Enter the FQDN for accessing the vFunction dashboard.
        Use a domain name and not an IP address.
        Note: To use TLS, enter "https://", and make sure you have the certifications and key. Otherwise enter "http://" (this exposes the server through HTTP).
      - **org_name**: Enter your organization name.
      - **upgrade**: Enter your preferred auto-upgrade mode:
        - **Daily** - The operator will check for a new version every day at 3 AM and will install it automatically (default).

    - **Always** - The operator will check for a new version every 10 minutes and will install it automatically.
    - **Never -** Do not upgrade automatically.
- **admin.email**: Enter the email address of your administrator.
- **admin.name**: Enter the name of your administrator.
- **admin.password**: Enter the password you want to use for the vFunction administrator.
  The password should be at least 8 characters long, and consist of at least one lowercase letter, at least one uppercase letter, at least one number, and at least one special character.
- **smtp.user**: Enter the email address for a designated user for the SMTP server (optional).
- **smtp.password**: Enter this user's password (optional).
- **smtp.identity**: Enter the SMTP server identity (optional).
- **smtp.url**: Enter the SMTP server URL (optional).
- **tls.crt**: If your host FQDN starts with "https", paste in the certifications you have for using the TLS connection.
- **tls.key**: If your host FQDN starts with "https", enter the key you have for using the TLS connection.

c. (Recommended) - validate your yaml file with a yaml validator tool.

2. In OpenShift, create and open your project.
3. Go to **Operators** > **OperatorHub**, and search for **vFunction**.
4. Click the **vFunction Operator** box, and then click **Install**.
5. In the window that appears, click **Subscribe** to install the latest stable version of the operator.
6. In the **Installed Operators** window, select and click **vFunction Operator**, and then click **Create Instance**.
7. In the **Create vFunctionServer** window, paste the yaml file that you created in step 1.
8. Click **Create**.

You can now click on your new vFunction operand to see its details and installation progress.

Verify the installation

Check the operand installed successfully:

1. Check that the **Operand State** property shows **Working**.
2. Check that the **Successfully Installed** property shows **Yes**.

If **Operand State** shows **Failed**, it indicates the operand wasn't installed correctly. Check the events and logs for all pods (operator and image containers) for any issues. If you are unable to troubleshoot, contact vFunction support at support@vfunction.com.

After installation

The vFunction site is now accessible via the newly created vFunction application custom address, for example, **my.domain.com**. There are two ways you can access the vFunction dashboard:

Access using the router canonical hostname

1. Update your DNS provider by creating a canonical name (CNAME) record.
   This record should point to your host address, and to the "vfunction" subdomain of the OpenShift canonical router hostname as the alias.
   For example:
   my.domain.com. CNAME vfunction.apps.ocp4.my-openshift.com.
2. Find your cluster Router Canonical Hostname address in the newly created **vfunction-route-xxx** , located in your vFunction OpenShift project > **Networking** > **Routes** > **vfunction-route-xxx route** > **Router Canonical Hostname** field.
   You can now access the dashboard using your defined "host" spec property (as above).

Access using the nginx service location

Use this access method if your OpenShift is installed on a provider that supports exposing LoadBalancer-type services.

1. Update your DNS provider by creating a canonical name (CNAME) record.
   This record should point to your host address and to the **vfunction-nginx-xxx** service location.
   For example,
   my.domain.com. CNAME a05951ed7cdf-1394239323.us-east-1.elb.amazonaws.com.

2. Point your custom domain to the **vfunction-nginx-xxx** service's external IP location, which you can find in the OpenShift project > **Networking** > **Services** > **vfunction-nginx-xxx** service > **Service Address** > **Location** field.

Upgrade and rollback

The vFunction operator includes a built-in auto-upgrade mechanism.

Choose 1 of 3 auto-upgrade modes:

- **Daily -** The operator will check for a new version every day at 3 AM and will install it automatically (default).
- **Always** - The operator will check for a new version every 10 minutes and will install it automatically.
- **Never -** Do not auto-upgrade.

To upgrade on-demand, change the mode from **Never** to **Always**, and change it back to **Never** after the operand is upgraded.

During the upgrade, the **Operand State** property changes to **Upgrading...** and returns to **Working** after a successful upgrade.

If the upgrade fails, the operand is automatically rolled back to the last working version. The failed version is marked as defective, and the operator will not attempt to upgrade to it again.

Air gapped (offline) installation

vFunction provides an "air gapped" (offline) installation package for installing the vFunction server operator on restricted network environments. The package can be found in the vFunction portal.

The installation consist of 3 parts:

1. Downloading, extracting and setting up the scripts & manifests with the desired OpenShift project, local registry address and port.
2. On a *non*-restricted network environment, mirroring operator and container images into a local registry that is accessed by the restricted OpenShift cluster.
3. Installing the operator by applying all manifests on the restricted network OpenShift and creating a new vFunction operand.

vFunction

In order to install the air gapped vFunction operator please do as follows:

1. Verify you have a linux machine with “oc” client installed and connected to your OpenShift cluster. Also verify you have a linux machine with access to both your local registry and the RedHat registry (may be on the same machine as the client or a different one).
2. Download and extract the package (tar zxvf) and go to the extracted “vfunction-operator-offline-installation” folder.
3. Run *bash init.sh -n <namespace> -h <local registry host name> -p <local registry host port>* to set the namespace (project) and local registry address and port.
4. Mirror all participant containers to a local registry by running *bash online/mirror-images.sh*. (This should be done on the machine with access to both the local and RedHat registry. You may copy the stand-alone mirror-images.sh script to a different machine and run it there, if needed).
5. Apply all manifests on the restricted network OpenShift cluster by running *oc apply -f offline/*
6. You should now see the vFunction operator installed in the **Operators -> Installed Operators** page in the OpenShift console.
7. Continue with installing the vFunction operand the same as described above for a regular installation, but make sure you add a new field *“offline”: “Yes”* to the operand CR yaml file.

In order to update the vFunction operand version on air gapped environments, when a new offline installation package is available, you should download the new package to a clean folder and follow steps 1 to 6 as described above.
Please note that step 7 is not necessary at this stage because the operand is already in place. Furthermore, there is no need to change anything in its CR yaml file - the new operator will upgrade to the new operand version as soon as it is installed.