Development Tools in Management VPC

The intent of this demo is to show how you can interact with the management cluster, and to introduce the tools used for a Cloud Native development workflow using the Cloud Native Toolkit. This will provide a very basic introduction to each of tools and show the user interfaces so that they will be more familiar in subsequent presentations.

Goals for the Demo:

- Familiarize the audience with tools used with the Cloud Native Toolkit

Prerequisites:

- If you have not already done so, request access to the FS Cloud demo environment at: https://techzone.ibm.com/collection/ibm-cloud-for-financial-services

- Download the OpenVPN client
  - Windows https://openvpn.net/community-downloads/
  - MacOS https://openvpn.net/client-connect-vpn-for-mac-os/
  - Linux https://openvpn.net/download-open-vpn/

- Download the techzone.ovpn VPN certificate and add it to the OpenVPN client
  - Link https://techzone-iam-agent.eqtyaj6hk2k.eu-de.codeengine.appdomain.cloud/vpn/download

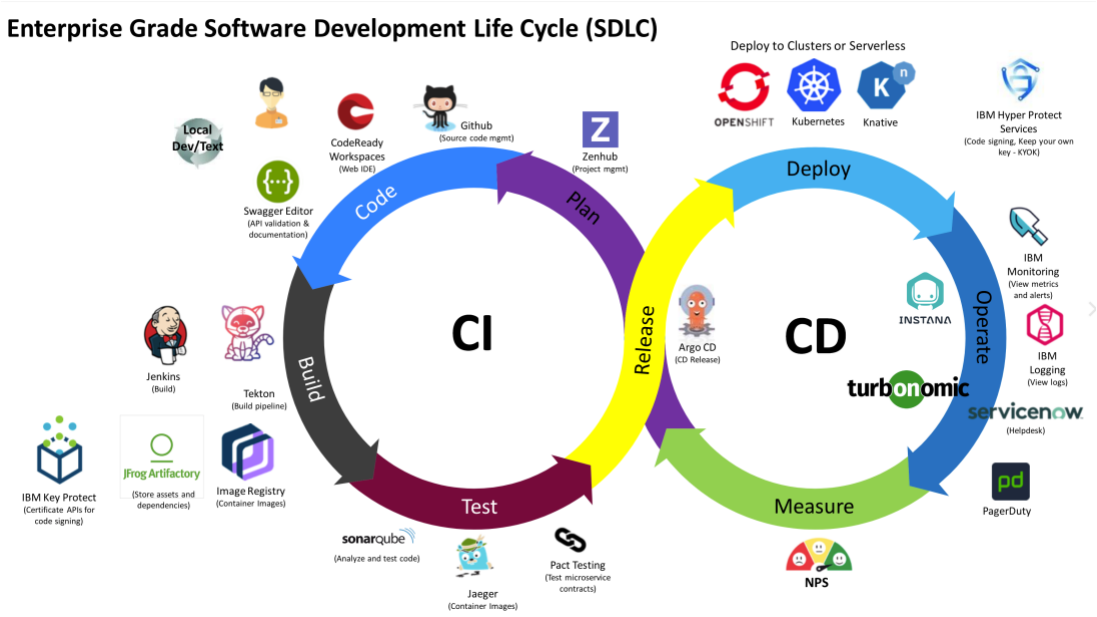
Demo Steps:

1. Set the stage… the tools that we're about to show are part of an enterprise-grade software development life cycle. This is the iterative process that enterprises use to build and deliver software reliably and consistently.

   It covers everything in the software development lifecycle, including continuous integration (CI) and continuous delivery (CD) phases.

   The continuous integration phase covers source control management, automated building, automated testing, container image and artifact management, and inspection/analysis for code quality and vulnerabilities.

   The continuous delivery phase covers the cycle of delivering those container images and artifacts into production, monitoring and measuring performance, and using this cycle to feed improvements and new requirements back into the continuous integration cycle to improve the overall solution.

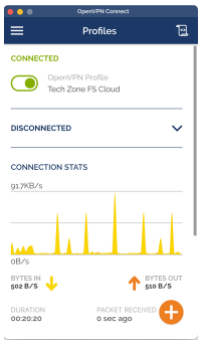


   The tools that we are using for an enterprise software development lifecycle (and are covered in this document) are components of the Cloud Native Toolkit (https://cloudnativetoolkit.dev/)

2. Now, let's examine how what tools we use to deliver software with this lifecycle within our FS Cloud instance.

3. First, we need to connect to the management cluster.

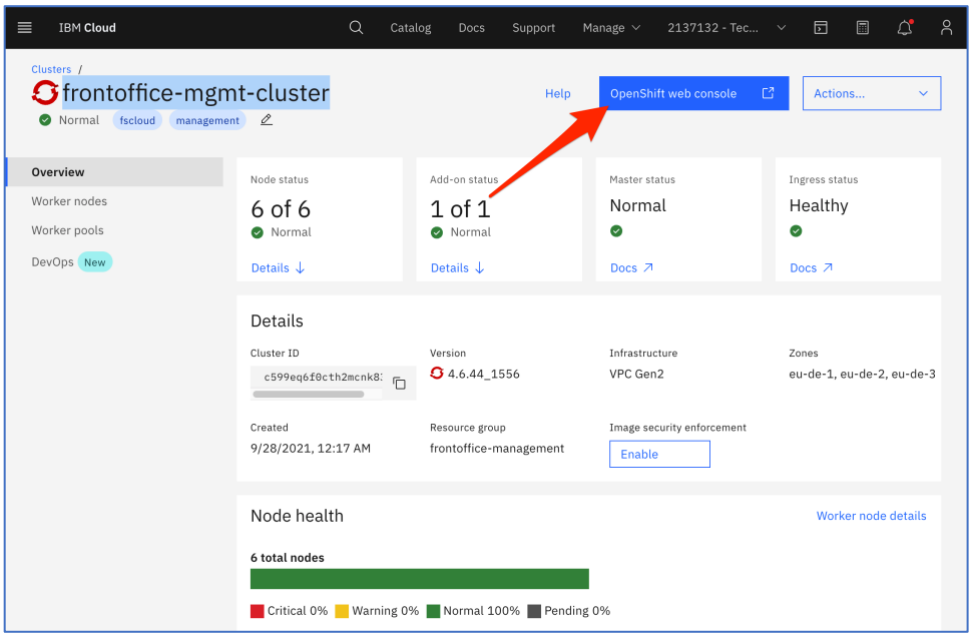
    a. Connect the OpenVPN Client with the Tech Zone demo profile



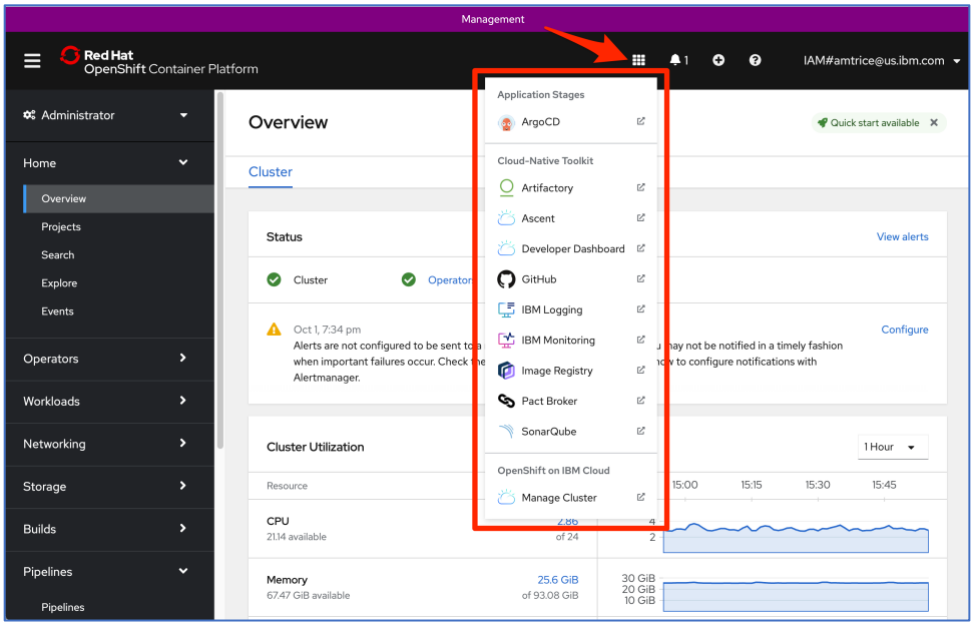
    b. Navigate to https://cloud.ibm.com/kubernetes/clusters

    c. Select the "**frontoffice-mgmt-cluster**" instance to view the cluster details

    d. Click the "OpenShift web console" button to bring up the OpenShift Dashboard



4. In the OpenShift Dashboard, click on the Application Launcher menu.



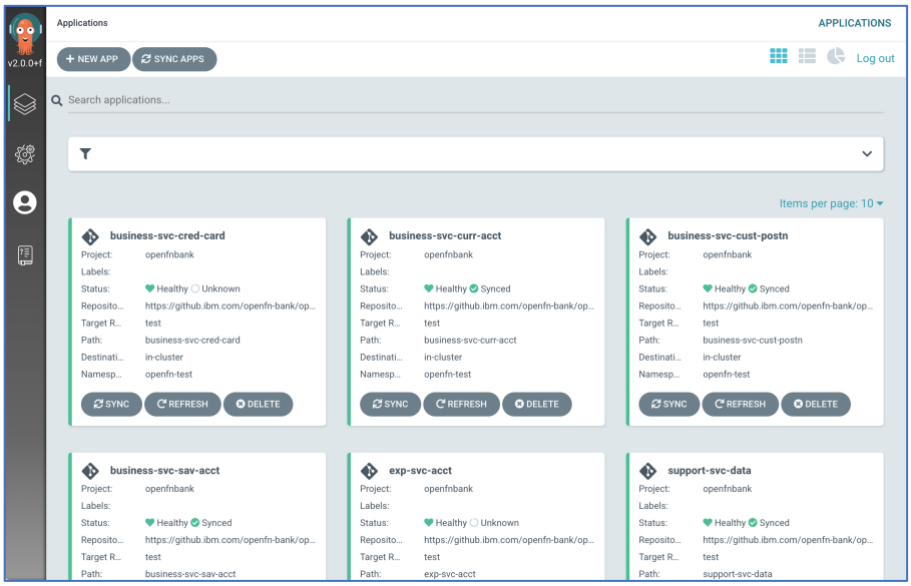
Here we want to highlight all of the tools we're going to cover and are instrumental in a an enterprise software development lifecycle:

a. **ArgoCD** - Continuous Delivery via GitOps. ArgoCD is used to deliver applications into the test and production environments.
b. **Artifactory** - managing assets/build artifacts. We will use Artifactory to store versioned artifacts that are produced by pipelines that automatically build the services for the OpenFN banking application.
c. **Ascent** – View and download reference architectures and terraform templates for FS Cloud Architectures
d. **Developer Dashboard** - get code starter kits and learning resources
e. **Github** – link to Cloud Native Toolkit GitHub org. This is where you can find source code to pipeline tasks and terraform modules used within the Cloud Native Toolkit
f. **IBM Logging** – link to IBM Cloud Logging. This is used to store and manage application logs (part of the continuous integration lifecycle)
g. **IBM Monitoring** – link to IBM Cloud Monitoring. This is used to manage cluster and resource performance and load (part of the continuous integration lifecycle)
h. **Image Registry** – link to IBM Container Registry. This is used to store container images that are produced by pipelines that automatically build the services for the OpenFN banking application.
i. **Pact Broker** – contract testing using Pact. This is used to manage and test software contracts for services that are part of your application. This is used to prevent breaking changes in your service contracts.

j. **SonarQube** – code quality analysis. This is used to scan your code for coding quality and vulnerabilities.

5. Now let's look a bit closer at each one of these tools.

6. In the application launcher menu, click on the "ArgoCD" link. You'll be taken to the ArgoCD login screen. Click on the "Log in via OpenShift " button to log in.

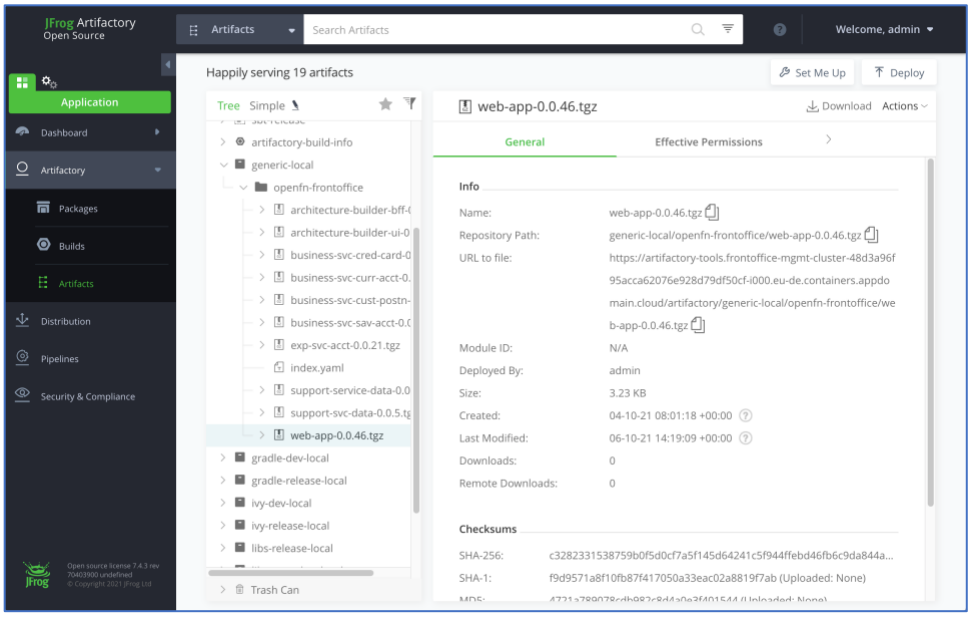


Once logged in you can see the ArgoCD dashboard. ArgoCD is used to deliver applications into the test and production environments.

This screen shows all of the GitOps continuous delivery deployments configured to deploy into this cluster and their status/health. We'll go into greater detail on GitOps workflows later.

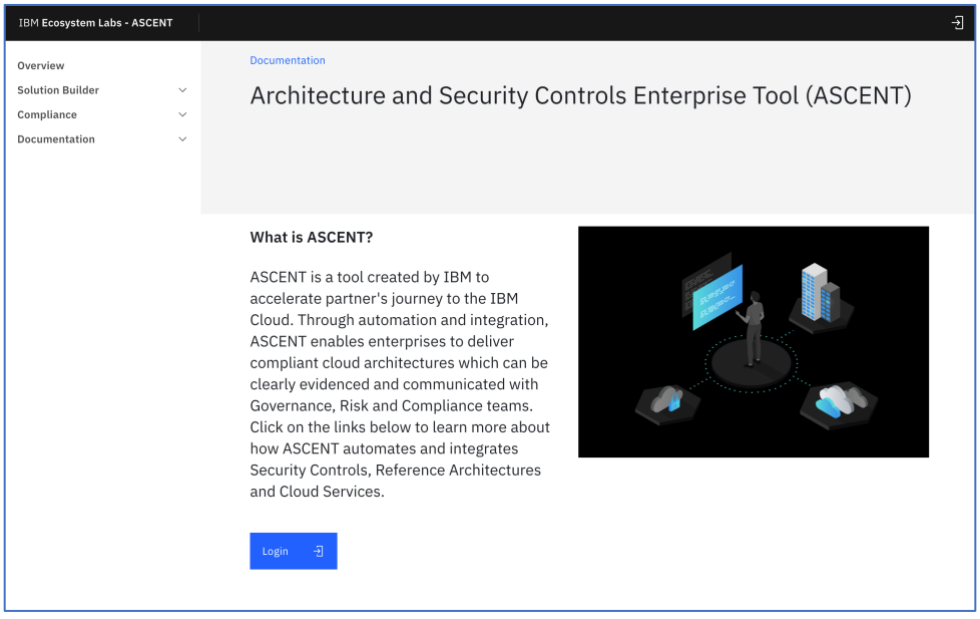
7. Go back to the OpenShift Dashboard, open the Application Launcher menu, and click on the "Artifactory" link. Artifactory is a universal artifact repository. It is used to store assets for your applications. In this case we're using it to store versioned copies of files generated by our DevOps build processes.

Click on "Artifacts" in the left menu, and browse the tree structure. Assets generated by pipelines in this cluster are placed in generic-local/openfn-frontoffice/



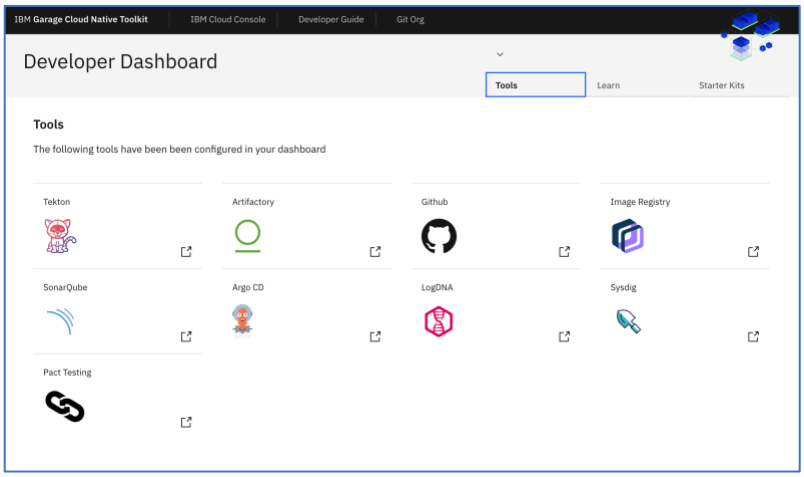
Using a centralized artifact storage solution makes it easier to track and search for files or artifacts, and allows you to keep versioned copies of all the artifacts of your applications. This makes it easy to reproduce specific build versions of your solutions, and makes it easy to roll back to one of those specific versions.

8. Go back to the OpenShift Dashboard, open the Application Launcher menu, and click on the “Ascent” link. Ascent enables you to access and download terraform templates for IBM reference architectures.

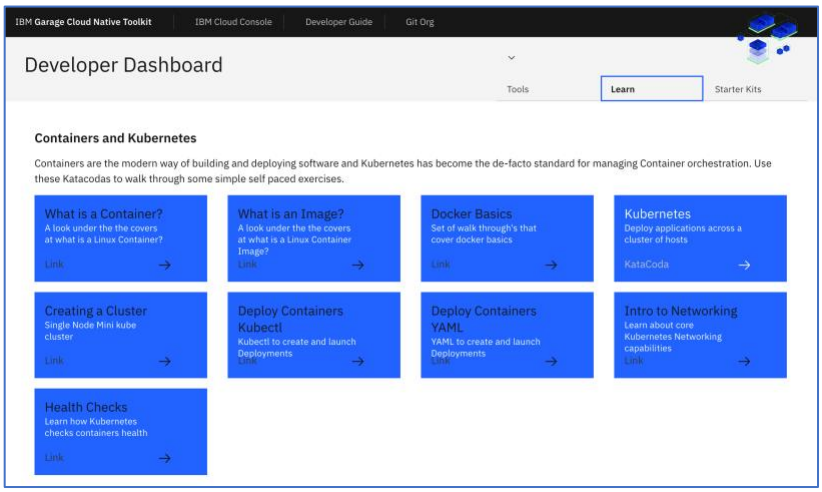


The Ascent tool is used to view FS Cloud reference architectures, download terraform templates for producing FS Cloud architecture instances, and map compliance controls to specific resource instances.

9. Go back to the OpenShift Dashboard, open the Application Launcher menu, and click on the "Developer Dashboard" link. The Developer Dashboard shows links to all the tools associated with this cluster, plus learning resources and application source code starter kits.

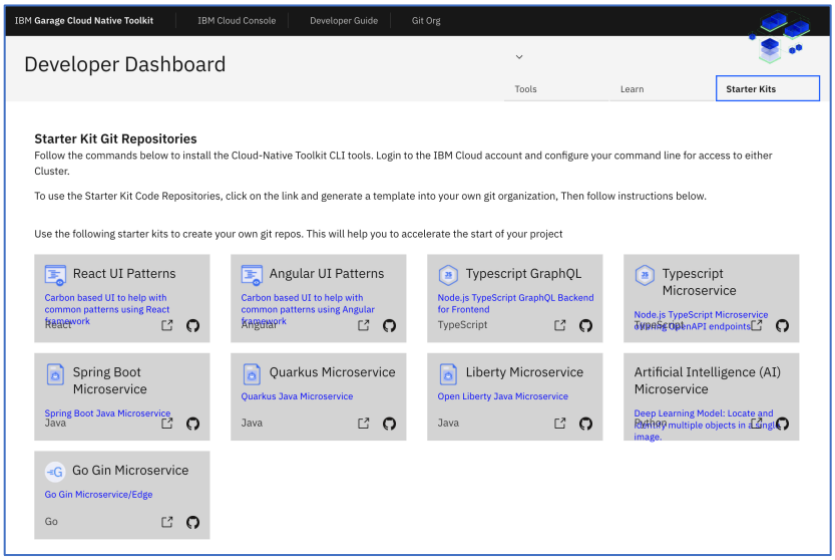


Click on the "Learn" tab to see links to various learning resources. This includes links for Cloud Native development learning and best practices.



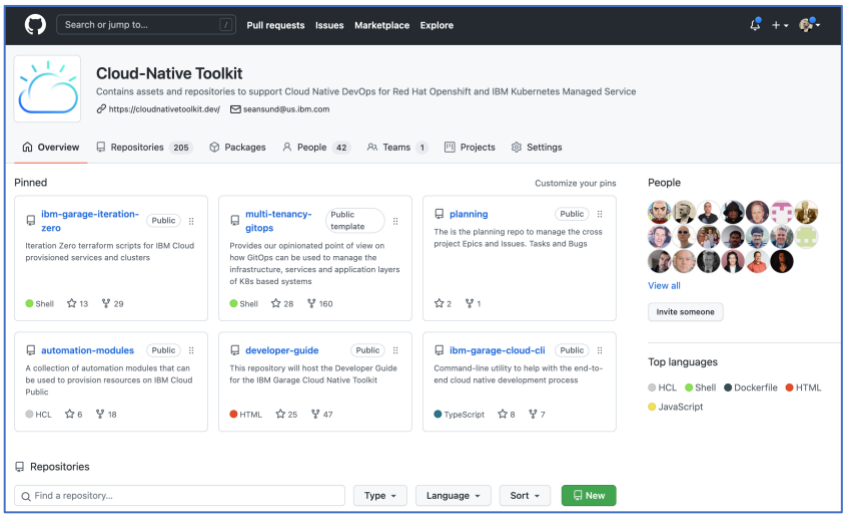
Click on the "Starter Kits" tab to see a list of application starter kits. These are fully functional production-grade code starters that can be cloned to get started quickly and

easily on a new project.



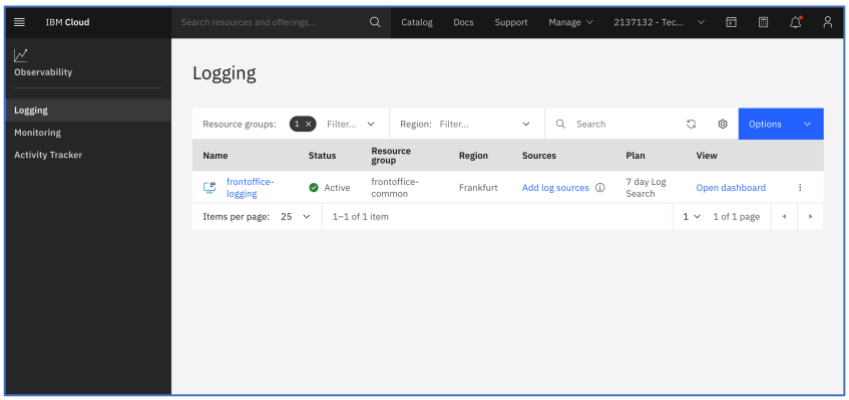
10. Go back to the OpenShift Dashboard, open the Application Launcher menu, and click on the “GitHub” link. This will open the https://github.com/cloud-native-toolkit GitHub org.

    Here you can find application source code, terraform templates, and definitions for Tekton pipeline tasks that are used by the Cloud Native Toolkit.

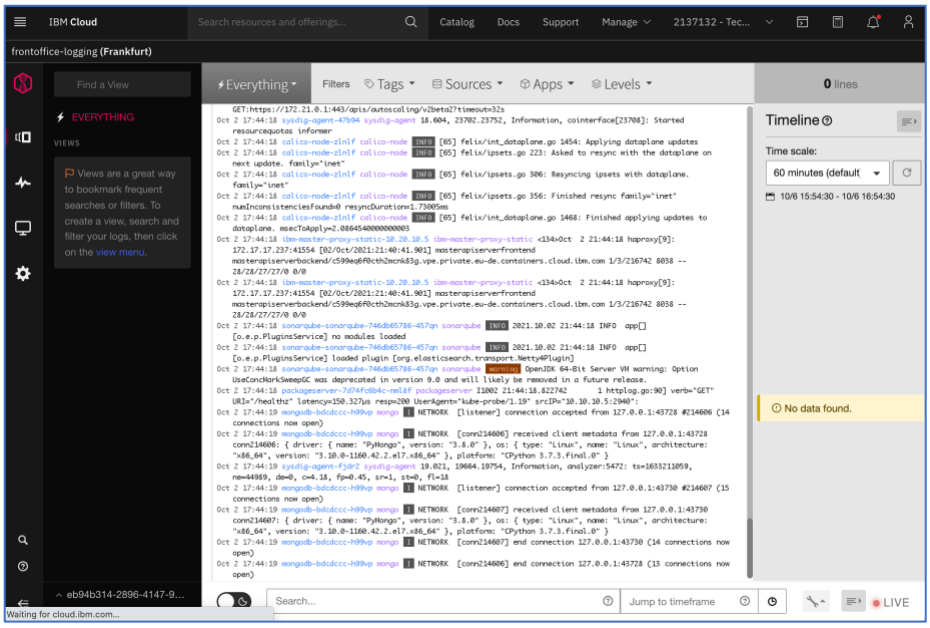


11. Go back to the OpenShift Dashboard, open the Application Launcher menu, and click on the “IBM Logging” link.

This will take you to https://cloud.ibm.com/observe/logging, where you can see your logging instances for the FS Cloud architecture.

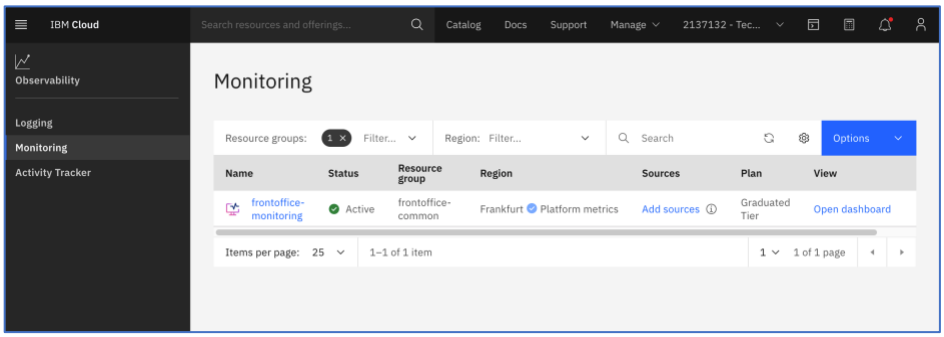


Click on the “Open Dashboard” link for the frontoffice-logging instance to open the log dashboard. Here you can see all of the logs for this instance.



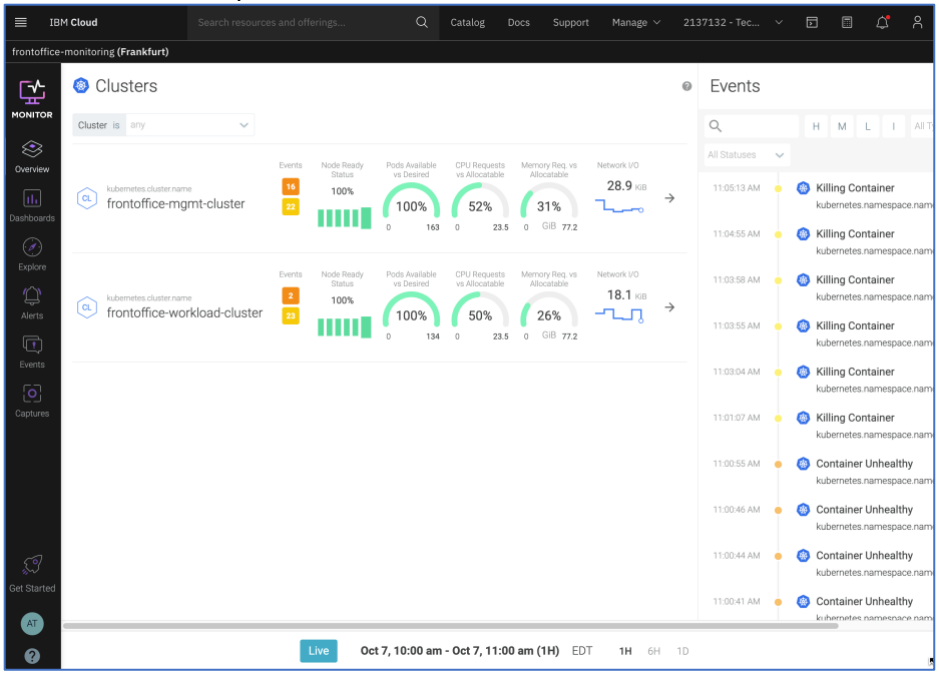
This allows you to view application logs, track interactions, and identify issues within your applications running in the FS Cloud demo architecture instance.

12. Go back to the OpenShift Dashboard, open the Application Launcher menu, and click on the “Monitoring” link. This opens https://cloud.ibm.com/observe/monitoring, where you can access a Sysdig instance to monitor the status and performance of your Kubernetes clusters.



Click on the “Open Dashboard” link to access the Sysdig dashboard.

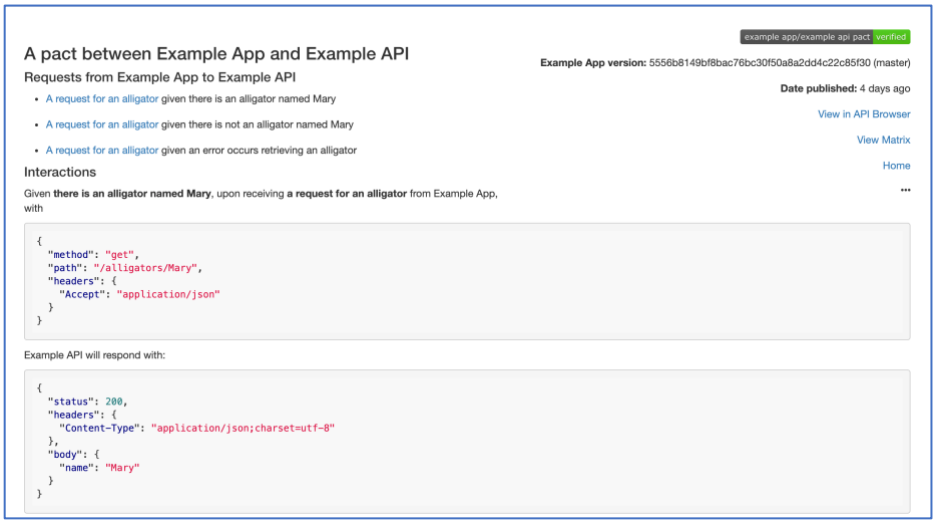
Once the dashboard loads, click on the “Overview” tab, and select “Clusters” to see an overview of the OpenShift clusters for the FS Cloud reference architecture instance.



Sysdig allows you to view the performance of your resource instances. Here we can use it to monitor the status of our management and workload OpenShift clusters. This can tell us the load (what percent of resources are being utilized), network traffic, and details about general performance and resource consumption for our cluster and workloads.

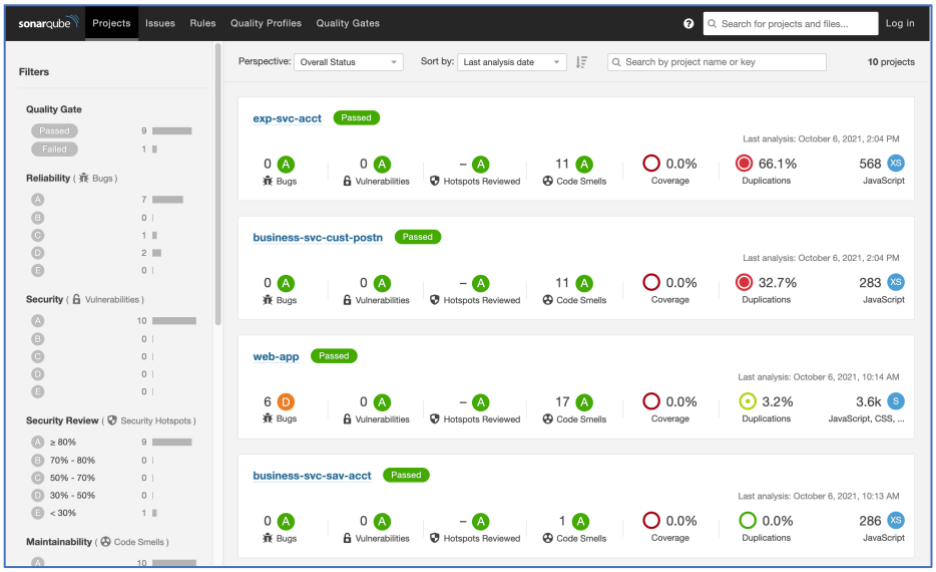
13. Go back to the OpenShift Dashboard, open the Application Launcher menu, and click on the “Pact Broker” link. This will open up the Pact Broker user interface, which can be use to test API contracts within your applications.

The Pact Broker application can help you know which microservices within your application must be released together, and can help prevent breaking changes when releasing new versions of your applications.

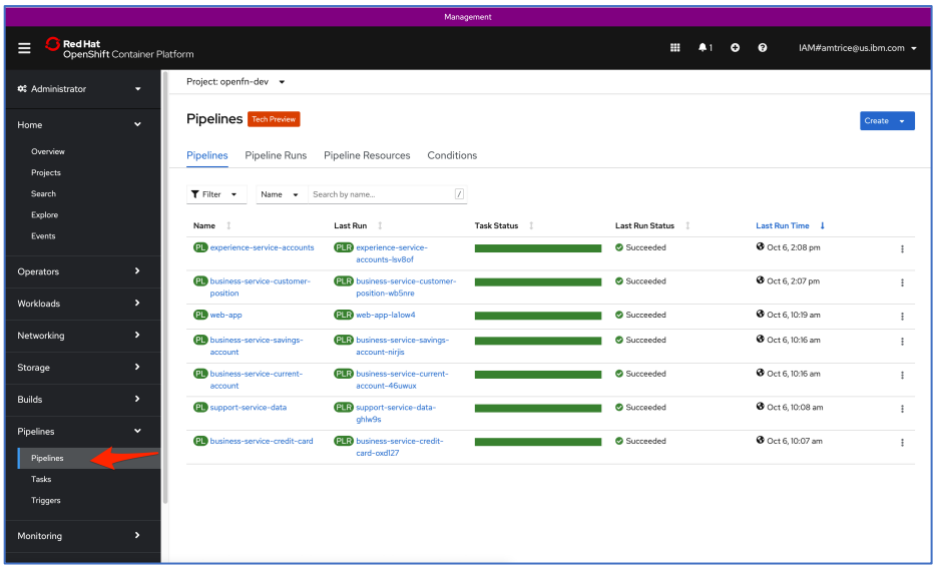


14. Go back to the OpenShift Dashboard, open the Application Launcher menu, and click on the “SonarQube” link. SonarQube is a code analysis tool. It will scan your codebase and provide feedback and analysis on coding practices, style, and raise known issues or vulnerabbilities.

    Click on the “Projects” tab in the header. Here you can see the analysis for the microservices used by the OpenFN banking application.



15. In the OpenShift Dashboard, click on the “Pipelines” link in the left menu. Emphasize this is where you find your DevOps Pipelines (based on Tekton), that you can use for automating your entire development (continuous integration) workflow (which we’ll cover in more detail in another demo).



16. Finally, navigate to https://cloudnativetoolkit.dev/ and reiterate that we’re using the development tools and templates that are available through the Cloud Native Toolkit.

The Cloud-Native Toolkit is an open-source collection of assets that enable application development and support teams to deliver business value quickly using Red Hat OpenShift or IBM Cloud-managed Kubernetes. This guide provides information to help Developers, Administrators, and Site Reliability Engineers use the Toolkit to support delivering business applications through the entire Software Development Life Cycle (SDLC).

Expand the “Reference” menu item, and show that there is additional detail available for every tool examined in the FS Cloud demo instance.