Hands On Workshop : Cloud

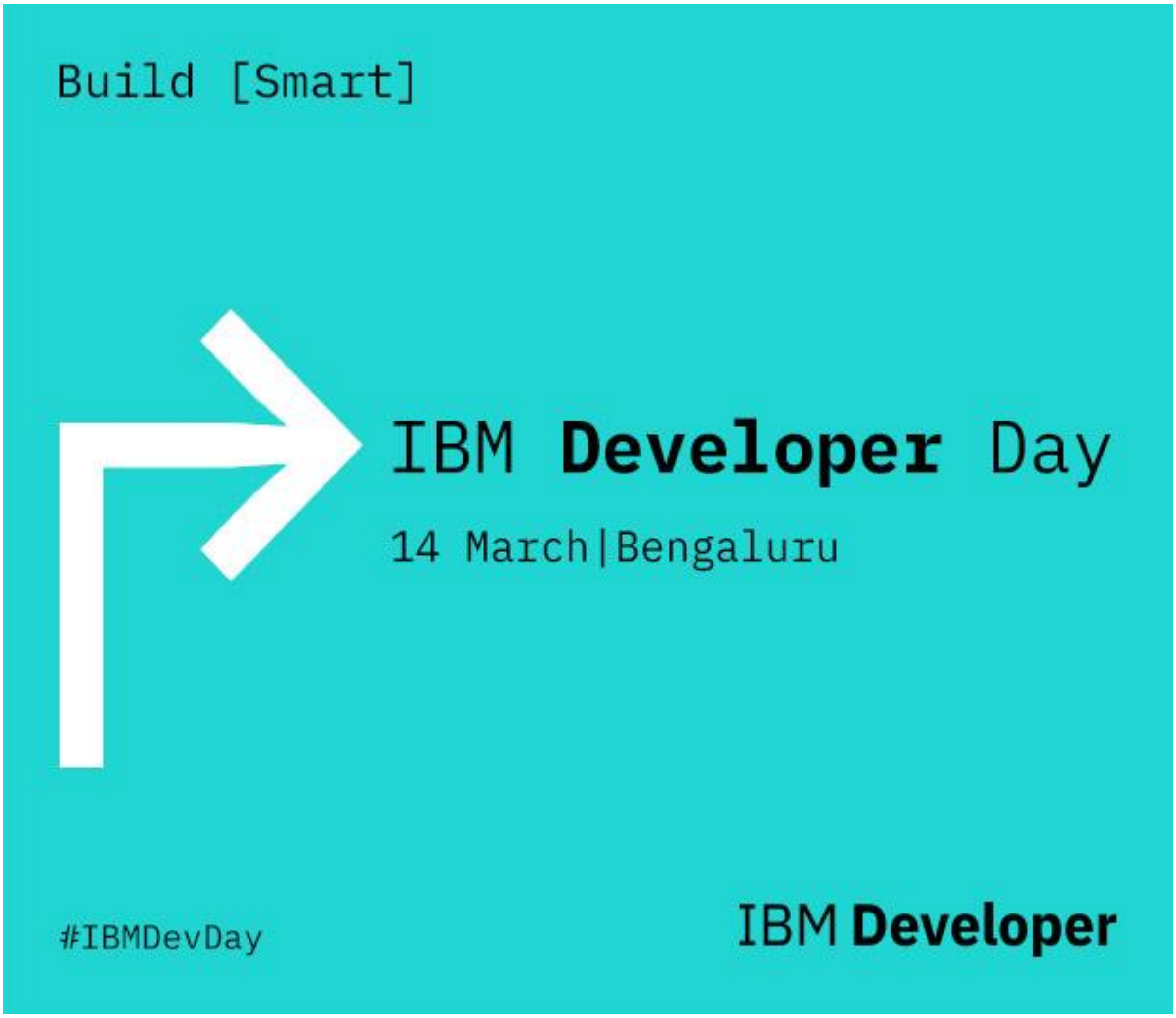


Table of Contents

Intro: Getting started - overview of the lab exercise

Microservices, APIs, and Cloud native architectures are hot technical topics in IT industry. In this hands-on lab, you will learn deploy microservices in containers in the cloud, how you can secure and monitor it.

You will be using a ToolChain (DevOps) capability available with IBM Cloud to deploy these microservices on IBM Cloud Kubernetes Managed Service

The primary technologies covered in the lab include: Docker, Node.js , Reddis Db, Kubernetes((BM) .

What you need:

1. **Laptop** – Install IBM Cloud Kubernetes Service plug-in CLI , install the **IBM Cloud CLI**
   Note: Google Chrome or Firefox browser is recommended
2. **IBM Cloud Account** – IBM Cloud required for Deploying microservice to a IBM Kubernetes Cluster Service
3. **Internet Connectivity**

IBM Cloud Kubernetes Free Cluster

This will take 20 -25 mins to spin up your cluster, which will be used to deploy your application later in the lab.

Refer this

https://console.bluemix.net/docs/containers/container_index.html#container_index

Or You can follow below steps

Login to your IBM Cloud Account and Create a Kubernetes Free cluster (Using GUI) Steps Below.

- Login to IBM Cloud with your credentials.
- Go to Catalog
- On Left hand side, Select container and click on Kubernetes Service
- Click on Create
- Under the cluster type. Select Free cluster.

On Geography select South America.

Give a Unique name to your Cluster and select Create Cluster.

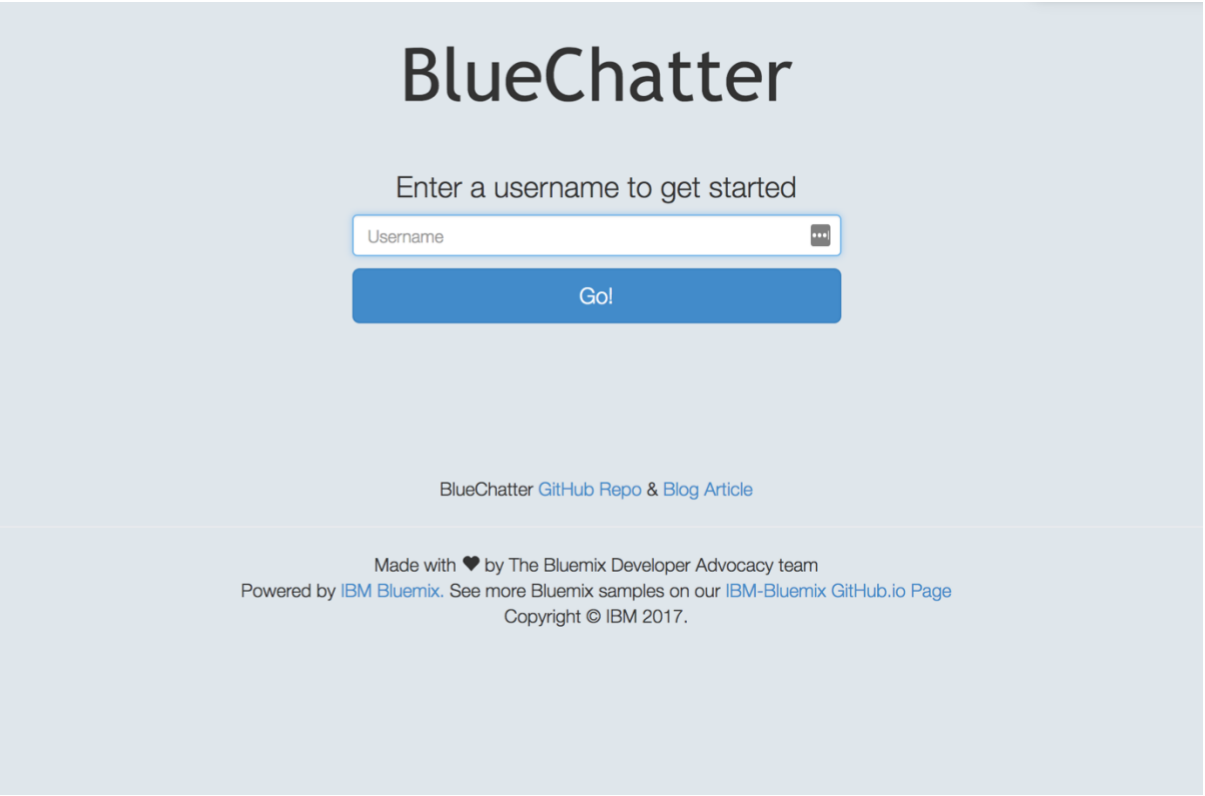
(This will take 20 -25 mins to spin up your cluster, which will be used to deploy your application later in the lab below).

Lab Prerequisites:

Before completing this lab, you are required to complete the lab prerequisite steps. These steps will guide you through the setup of a IBM Cloud account, command line tools , and the creation of a Kubernetes Cluster. You can review the prerequisite instructions here. Note that you will NOT be able to complete the lab until you complete the prerequisites.

What you will be building:

In this lab you will be building BlueChatter Application as microservices using multiple technology stacks, and you will deploy and test these microservices on a IBM Cloud Kubernetes Managed Cluster. The diagram shows the application that are covered:



BlueChatter

Enter a username to get started

BlueChatter GitHub Repo & Blog Article

Made with ♥ by The Bluemix Developer Advocacy team
Powered by IBM Bluemix. See more Bluemix samples on our IBM-Bluemix GitHub.io Page
Copyright © IBM 2017.

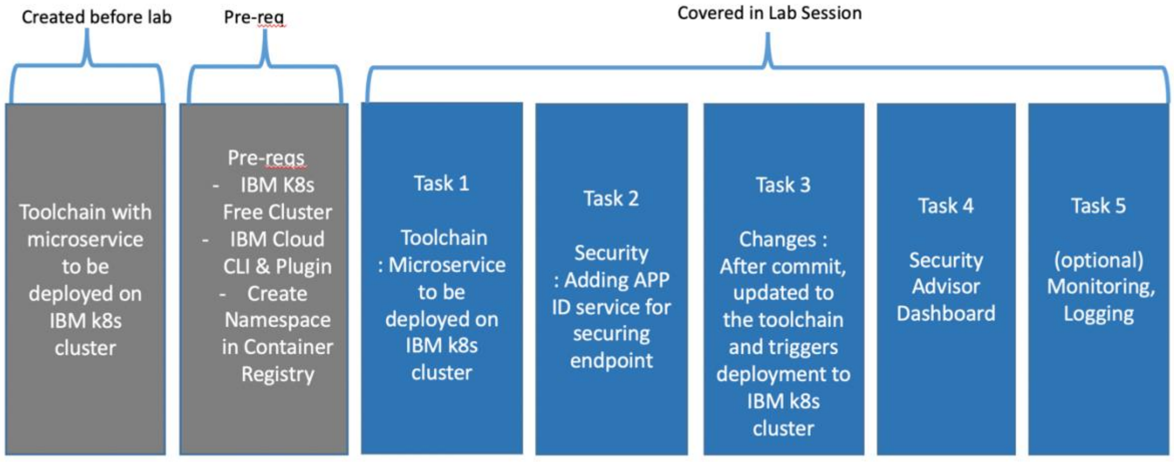
BlueChatter uses Node.js and Express for the server. On the frontend, BlueChatter uses Bootstrap and Jquery. The interesting part of this application is how the communication of messages is done. The application uses long polling to enable the clients (browsers) to listen for new messages. Once the app loads successfully, a client then issues a request to the server. The server waits to respond to the request until it receives a message. If no message is received from any of the chat participants, it responds back to the client with a 204 - no content. As soon as the client gets a response from the server, regardless of whether that response contains a message or not, the client will issue another request and the process continues.

Once the application is deployed through Toolchain, next you will secure this application by providing an authentication mechanism. For the authentication, you will use App ID service from IBM Cloud Catalog to add authentication to your web application and protect your APIs and back-ends running on IBM Cloud. Add email/password based sign-up and sign-in, and MFA with App ID's scalable user registry - Cloud Directory, or social log-in with Google or Facebook. For employee apps, use SAML 2.0 federation to let users sign-in with their enterprise credentials. For all app users, enrich their profiles with additional info so you can build engaging experiences.

Optionally, you will see how this deployment of k8s applications is monitored on IBM Cloud.

The components of the lab material are illustrated in the diagram below. These components include toolchain available on GITHUB URL as input for the lab, perquisite instructions to setup a development environment, and the five tasks that are covered in this lab handout.

An overview of our approach for the scoping, designing and modeling of microservices, deployment to IBM Cloud K8s Cluster , Security using APP ID and added monitoring will be provided during the lab session.


Created before lab
Pre-req
Covered in Lab Session
Toolchain with microservice to be deployed on IBM k8s cluster
Pre-reqs
- IBM K8s Free Cluster
- IBM Cloud CLI & Plugin
- Create Namespace in Container Registry
Task 1
Toolchain : Microservice to be deployed on IBM k8s cluster
Task 2
Security : Adding APP ID service for securing endpoint
Task 3
Changes : After commit, updated to the toolchain and triggers deployment to IBM k8s cluster
Task 4
Security Advisor Dashboard
Task 5
(optional) Monitoring, Logging

The remainder of this lab is divided into five tasks that provide hands-on experience with deploying microservices in the cloud using ToolChain. Each task builds on the activities of the previous one, so it is recommended to complete them in the order they are listed. The primary activities of each task are as follows:

Task 1:
You will be accessing the toolchain, as an input , to deploy the current version of BlueChatter application. This will ask your IBM Cloud Kubernetes Free Cluster details, and Container Registry details. Once you provide that, BlueChatter application will be deployed on your IBM Cloud Kubernetes Free Cluster

Task 2:
The Bluechatter Application is not secured after deployment from Task1. In this task2 , you will use App ID to add authentication to your application and protect your APIs and back-ends running on IBM Cloud. This changes will be done using ToolChain, and editing using Eclipse Orion Web IDE through the toolchain.

Task 3:
This new version of BlueChatter Application changes, now you will push the committed code changes by clicking the push button. This will trigger a new build in the tool chain. After successful build you will be asked to use this authentication method to access the application.

Task 4:
IBM Cloud Security Advisor is a security dashboard that provides centralized security management. The dashboard unifies vulnerability and network data as well as application and system findings from IBM Services, partners and user-defined sources. By centralizing visibility and enabling drill down to resolution, Security Advisor empowers the security admin to cohesively manage security on IBM Cloud workloads. You will understand this dashboard in this task.

Task 5:
(Optional)
This is an optional task, To understand more about the Kubernetes cluster and how to debug and get logs from your application.

(Optional)
Automatic Scaling and Manual Scaling on IBM Cloud Kubernetes Cluster.

Terminology:

ToolChain
DevOps toolchain that is preconfigured for continuous delivery with Vulnerability Advisor, source control, issue tracking, and online editing, and deployment to the IBM Cloud. In this lab, you will be using this for deployment to IBM Cloud (Kubernetes) Containers service.

Microservice
In short, the microservice architectural style is an approach to developing a single application as a suite of small services, each running in its own process and communicating with lightweight mechanisms, often an HTTP resource API. These services are built around business capabilities and independently deployable by fully automated deployment machinery. There is a bare minimum of centralized management of these services, which may be written in different programming languages and use different data storage technologies. For more information, visit https://martinfowler.com/articles/microservices.html

Container
A container image is a lightweight, stand-alone, executable package of a piece of software that includes everything needed to run it: code, runtime, system tools, system libraries, settings.
Kubernetes(K8s)

IBM K8s Free Cluster
Kubernetes is an open-source system for automating the deployment, scaling, and management of containerized applications. It consists of one or more Master nodes (machines) and one or more Worker nodes. The cluster is able to spread containers across the various nodes to balance resources (see Figure 1).

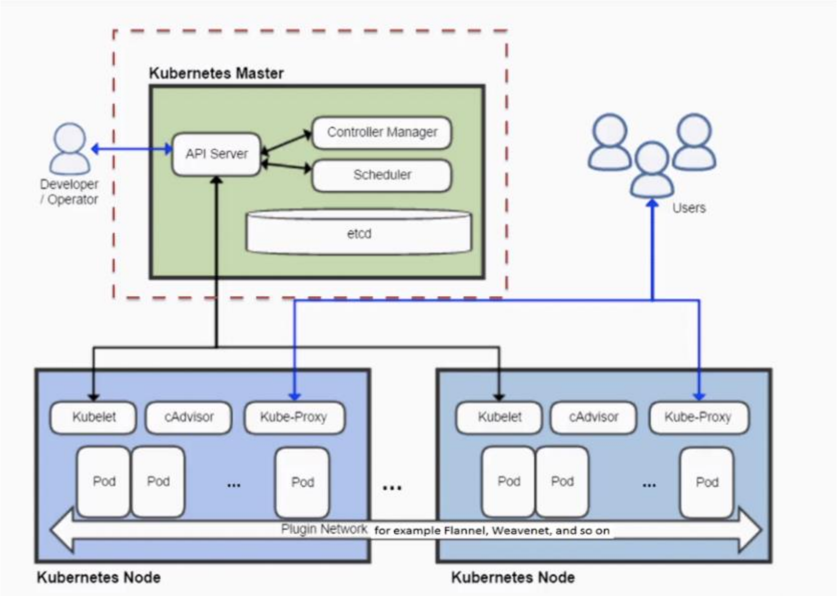


Figure 1: Kubernetes Architecture

Containers are deployed in Kubernetes Pods. Pods are logical grouping of containers which work closely together, sharing network and disk memory resources. Typically, a Pod includes a single container. Pods and their containers can communicate with other Pods and their containers by IP address. IP address are usually assigned dynamically every time a Pod is started or restarted, so keeping track of their IP addresses becomes impractical. To address this issue, Kubernetes Services are created (see Figure 2).

Security (App ID)

IBM Cloud App ID is the strategic identity service on IBM Cloud. Use it to add authentication to your mobile and web apps and protect your APIs and back-ends running on IBM Cloud. Enable email/password based sign-up and sign-in with Cloud Directory – App ID's scalable user registry, allow your employees to sign-in with their existing credentials via SAML federation, or use social sign-in with Facebook and Google. Host user profile info that you can use to build engaging experiences. IBM Cloud App ID is a cloud-native, managed service running in multiple regions and availability zones of IBM Cloud, providing data governance, access management, and other capabilities.

Task 1: Deploy BlueChatter (Microservices based) Application using ToolChain on IBM Public Cloud

Pre-req

You have read the 'Intro: Getting Started' section and completed the prerequisites listed there

This part of the lab explains how to complete the following tasks:
• Deploy ToolChain on IBM Cloud using your IBM Cloud Account

Estimated Duration : 7-10 Mins

Pre-req Checks:

1) *IBM Cloud Account Created*
2) *Cluster Created – in case Lite Account, please request PROMO Code, to create cluster in US South.*
3) *IBM CLI and Plugin installed*

(You will need to plugins for this workshop,

a) Container Service

b) Container Registry (in case not installed use below commands

```
- $ ibmcloud plugin install container-service
- $ ibmcloud plugin install container-registry
```

Overview

You will deploy BlueChatter application which consist of various micro services on IBM Cloud Kubernetes Free Cluster. You will use the tool chain capability available in ibm cloud.

1. Create Namespace for IBM® Cloud Container Registry

Warning: In case you already have namespace, you can skip creation of new space, and use the one which you have

Before you can store your Docker images in IBM® Cloud Container Registry, you must create a namespace.

Now you will create namespace.
To work with namespaces, install the container-registry CLI plug-in. (This was one the pre-req for workshop).

Follow below steps to create a namespace

- Login through the IBM Cloud CLI

```
$ ibmcloud login
```

- Login with Container Registry (optional)

```
$ ibmcloud cr login
```

- List available namespaces.

```
$ ibmcloud cr namespace-list
```

- In case namespace is available, you can use the same or create new using below command

```
$ ibmcloud cr namespace-add <your-namespace>
```

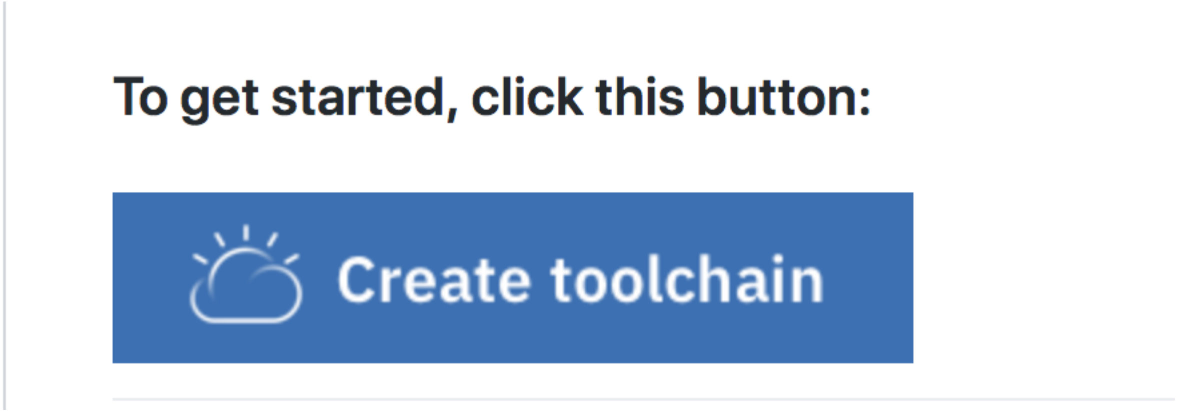
- Verify the namespace is created

```
$ ibmcloud cr namespace-list
```

2. Use Pre-configured Tool Chain

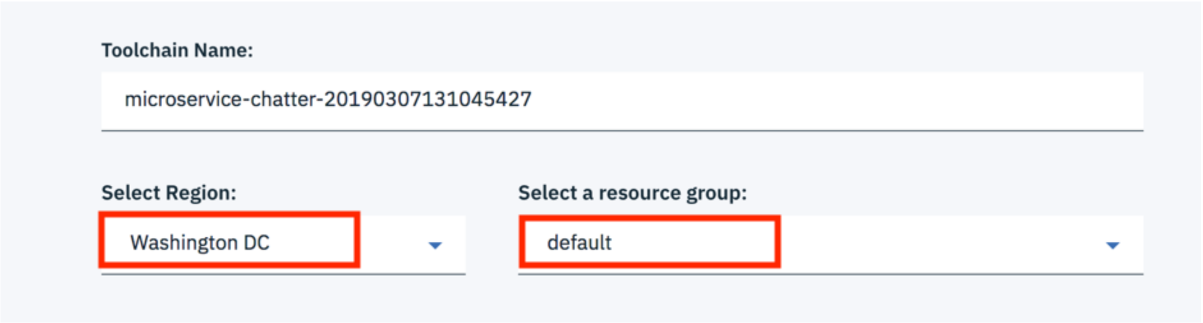
Open the git repo
>https://github.com/logeswtr/bluechatter

and go down and you will see “Create toolchain” icon like below, click on it



To get started, click this button:

The required fields will be populated with defaults. Make sure the **Region** is Washington DC or Dallas as below.

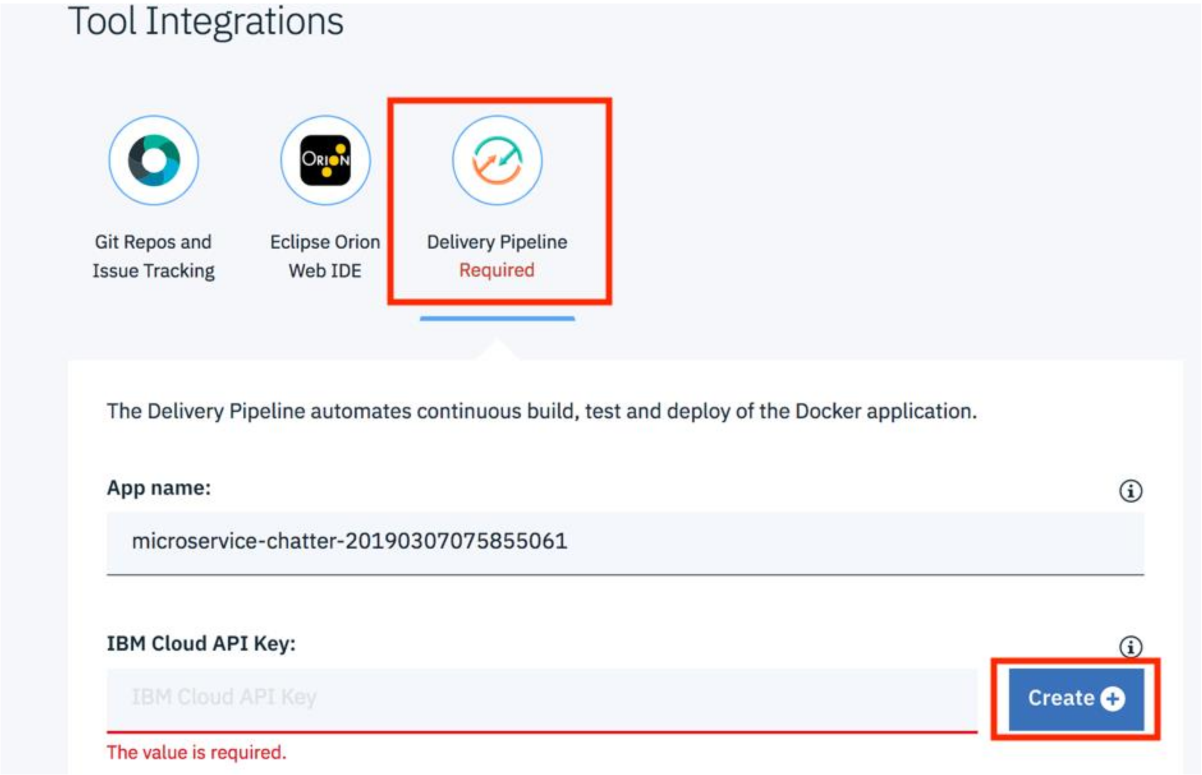


Toolchain Name: microservice-chatter-20190307131045427

Select Region: Washington DC

Select a resource group: default

Move to Delivery Pipeline tab in Tool Integrations section
Click create next to IBM Cloud API Key:



Tool Integrations

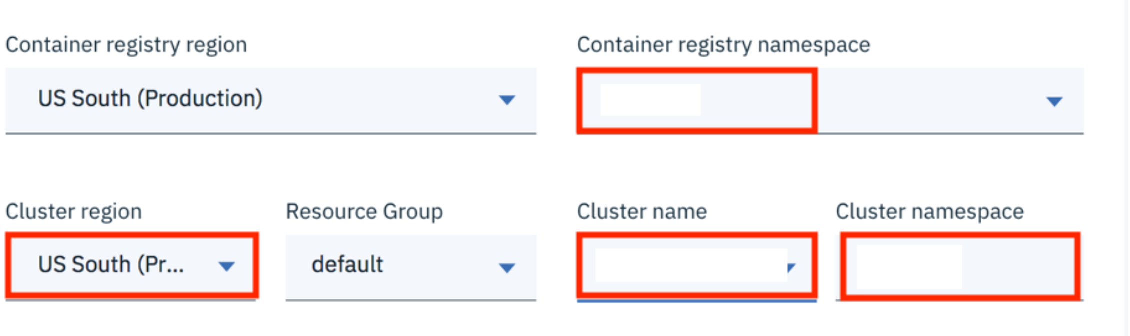
The Delivery Pipeline automates continuous build, test and deploy of the Docker application.

App name: microservice-chatter-20190307075855061

IBM Cloud API Key: IBM Cloud API Key

The value is required.

Make sure the rest of the sections are populated with appropriate values (like name of your cluster) as below.



Container registry region: US South (Production)

Container registry namespace:

Cluster region: US South (Pr...

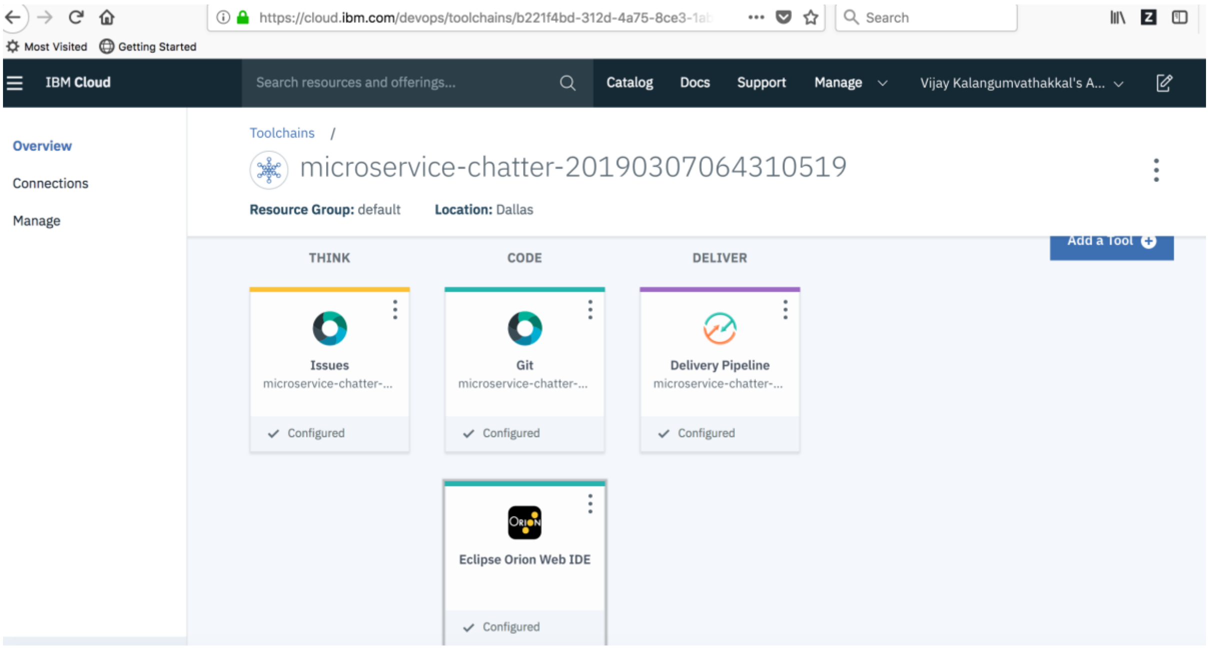
Resource Group: default

Cluster name:

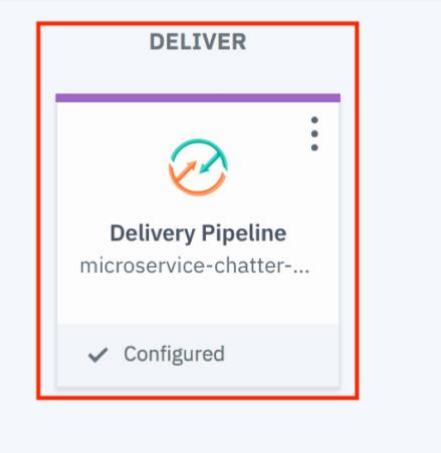
Cluster namespace:

Click create.

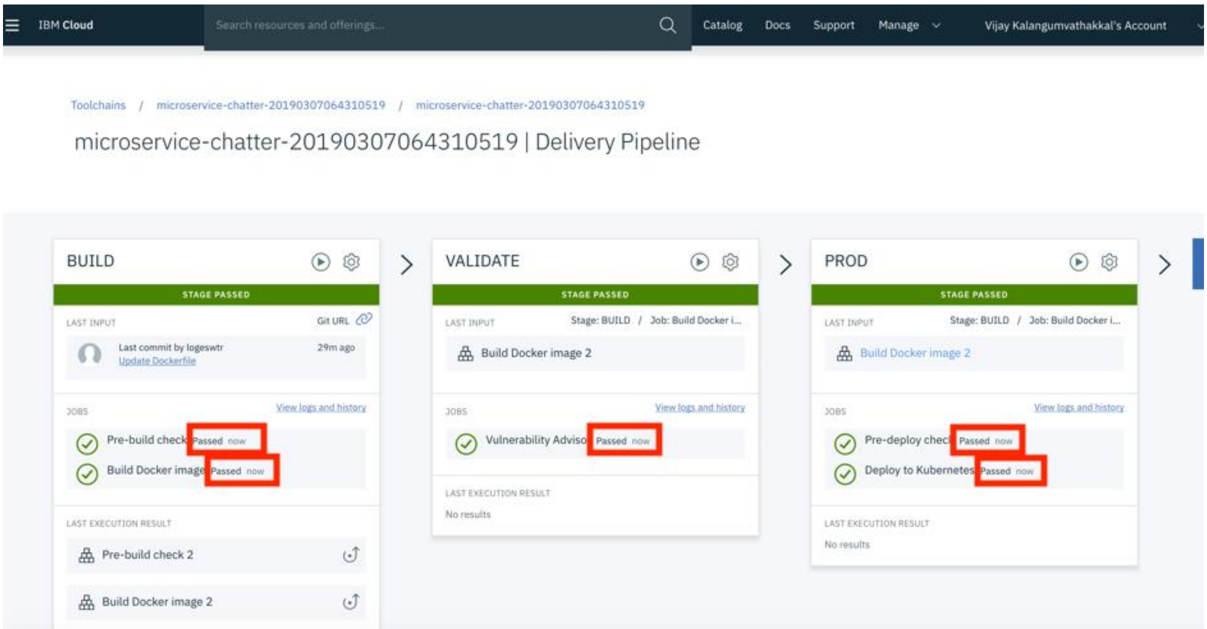
This will trigger the tool chain setup and looks like below.



Click on delivery pipeline



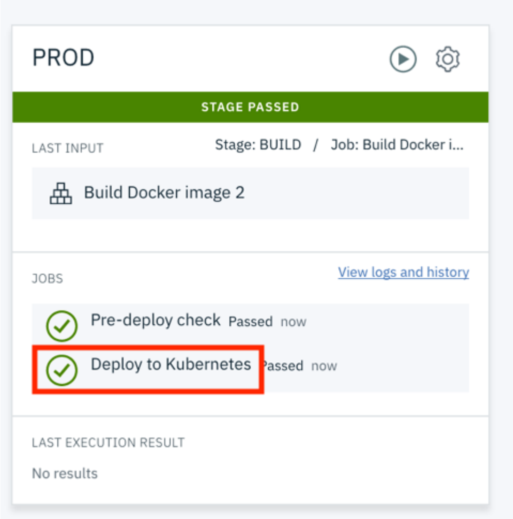
You can see various stages there like below. Please wait till all the 3 stages are in passed state.


IBM Cloud
Search resources and offerings...
Catalog
Docs
Support
Manage
Vijay Kalangumvathakkal's Account
Toolchains / microservice-chatter-20190307064310519 / microservice-chatter-20190307064310519
microservice-chatter-20190307064310519 | Delivery Pipeline
BUILD
STAGE PASSED
LAST INPUT
Git URL
Last commit by logeswtr
Update Dockerfile
29m ago
JOBS
View logs and history
Pre-build check Passed now
Build Docker image Passed now
LAST EXECUTION RESULT
Pre-build check 2
Build Docker image 2
VALIDATE
STAGE PASSED
LAST INPUT
Stage: BUILD / Job: Build Docker i...
Build Docker image 2
JOBS
View logs and history
Vulnerability Advisor Passed now
LAST EXECUTION RESULT
No results
PROD
STAGE PASSED
LAST INPUT
Stage: BUILD / Job: Build Docker i...
Build Docker image 2
JOBS
View logs and history
Pre-deploy check Passed now
Deploy to Kubernetes Passed now
LAST EXECUTION RESULT
No results

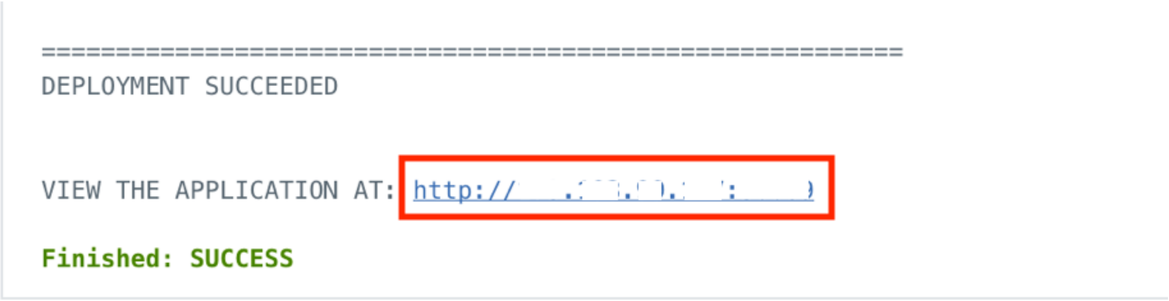
After this step, Following things happened,
1) The Application Code was pulled in, cloned to your repository.
2) IBM Cloud Tool Chain was triggered.
3) Container images of required micro services were built.
4) These images were checked for any vulnerabilities via IBM Vulnerability Advisor Service.
5) And the image is pushed Container registry and Deployed to IBM Kubernetes Free Cluster, you created via the tool chain.

3. Access the Deployed Application

Let’s now access the application.
Click on “**Deploy to Kubernetes**” in the prod stage in the pipeline



This will open the detailed page for this stage. Under logs section, scroll down to grab the deployed BlueChatter app link and hit it form a browser.

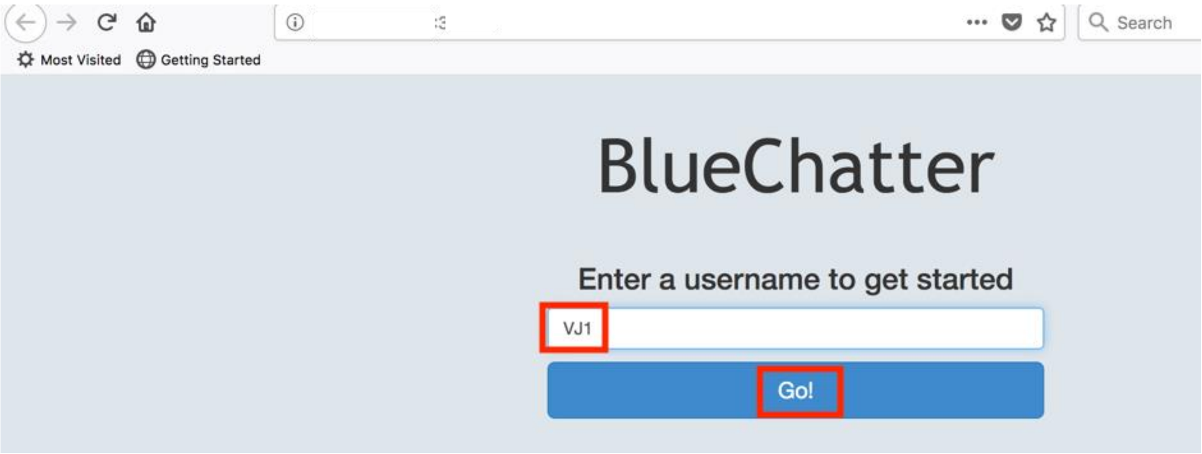


```
==========================================================
DEPLOYMENT SUCCEEDED

VIEW THE APPLICATION AT: http://[redacted]

Finished: SUCCESS
```

The BlueChatter app will be opened.

4. Test the Deployed Application

Enter a username and click Go

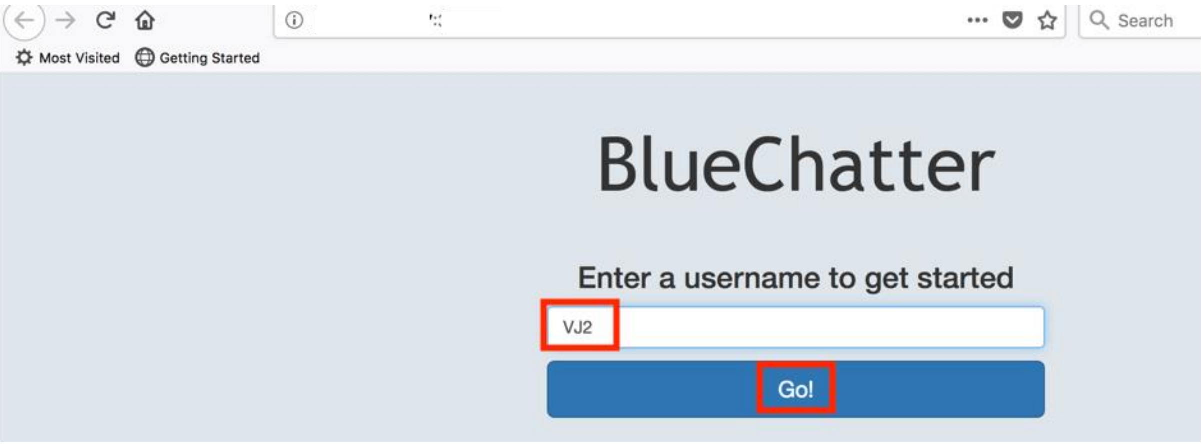


BlueChatter

Enter a username to get started

VJ1

Repeat the same step again with another user name



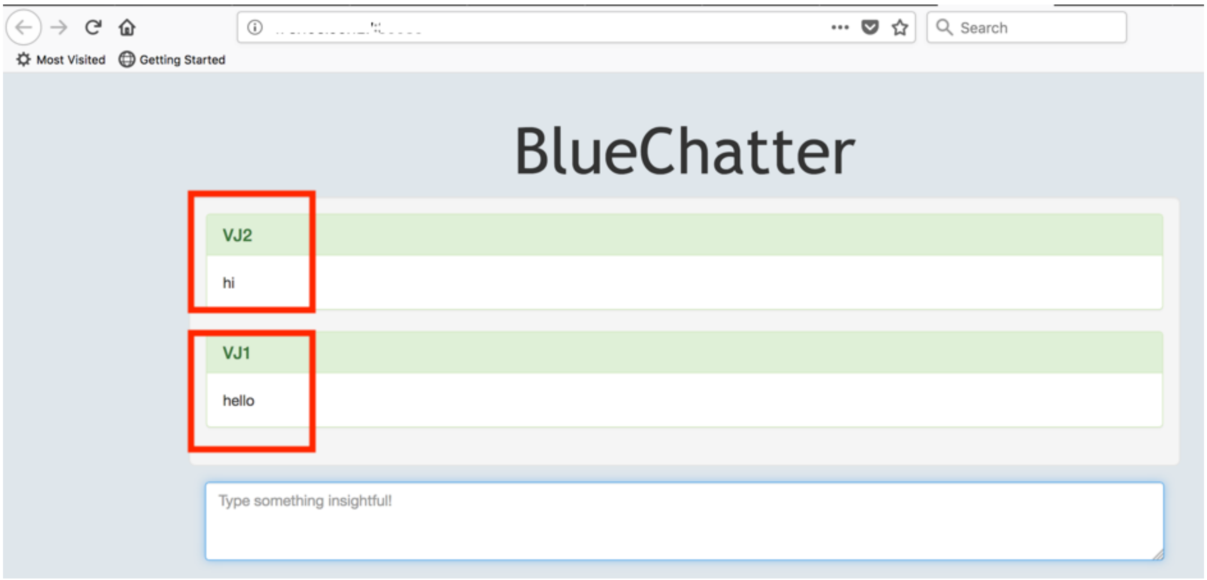
BlueChatter

Enter a username to get started

VJ2

Go!

This is simulating chat between 2 users VJ1 and VJ2. You can enter messages in any windows and vice versa and chat is in place.



BlueChatter

VJ2
hi

VJ1
hello

Type something insightful!

At this point we have successfully deployed and tested Application to IBM Kubernetes Free Cluster via IBM Tool chain.

Observe that there is no authentication or user engagement enabled for this website. You also notice that you can access the page without being authenticated.

Next You will secure this application by providing an authentication mechanism. For the authentication, we will use APP ID service from IBM Cloud Catalog.

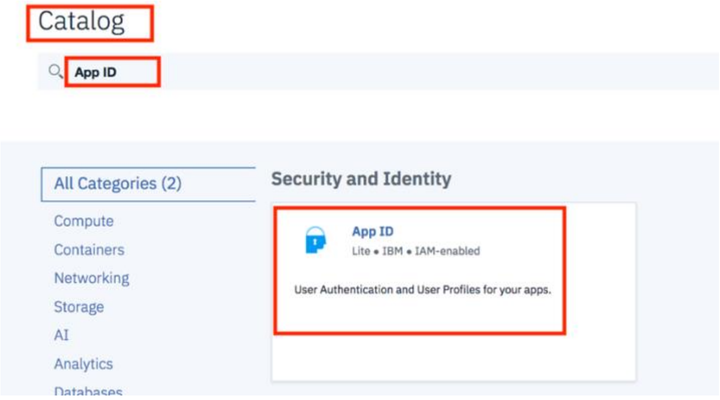
Task 2. Secure Application using App ID Service from IBM Cloud

You did deploy the "BlueChatter" Application which is not secured.

In this Task 2 , You will use App ID to add authentication to your application and protect your APIs and back-ends running on IBM Cloud. This changes will be done using ToolChain, and editing using **Eclipse Orion Web IDE** through the toolchain.

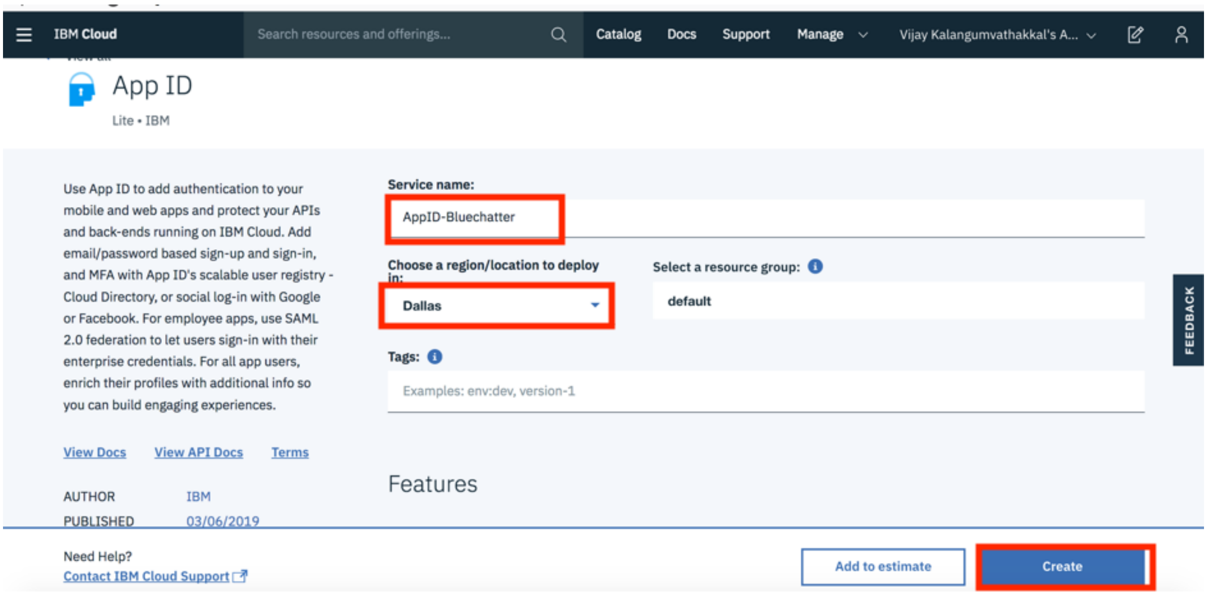
1. Create App ID service from IBM Cloud

Go to IBM Cloud Catalog and search for APP ID and create an instance with lite plan.



Please use the name as "AppID-Bluechatter". And Click on "Create"

Warning This is important , as later you will use this in the Kubernetes Deployment File.


IBM Cloud
Search resources and offerings...
Catalog
Docs
Support
Manage
Vijay Kalangumvathakkal's A...
FEEDBACK
Add to estimate
Create

App ID

Lite • IBM

Use App ID to add authentication to your mobile and web apps and protect your APIs and back-ends running on IBM Cloud. Add email/password based sign-up and sign-in, and MFA with App ID's scalable user registry - Cloud Directory, or social log-in with Google or Facebook. For employee apps, use SAML 2.0 federation to let users sign-in with their enterprise credentials. For all app users, enrich their profiles with additional info so you can build engaging experiences.

View Docs View API Docs Terms

AUTHOR IBM
PUBLISHED 03/06/2019

Need Help?
Contact IBM Cloud Support

Service name: AppID-Bluechatter

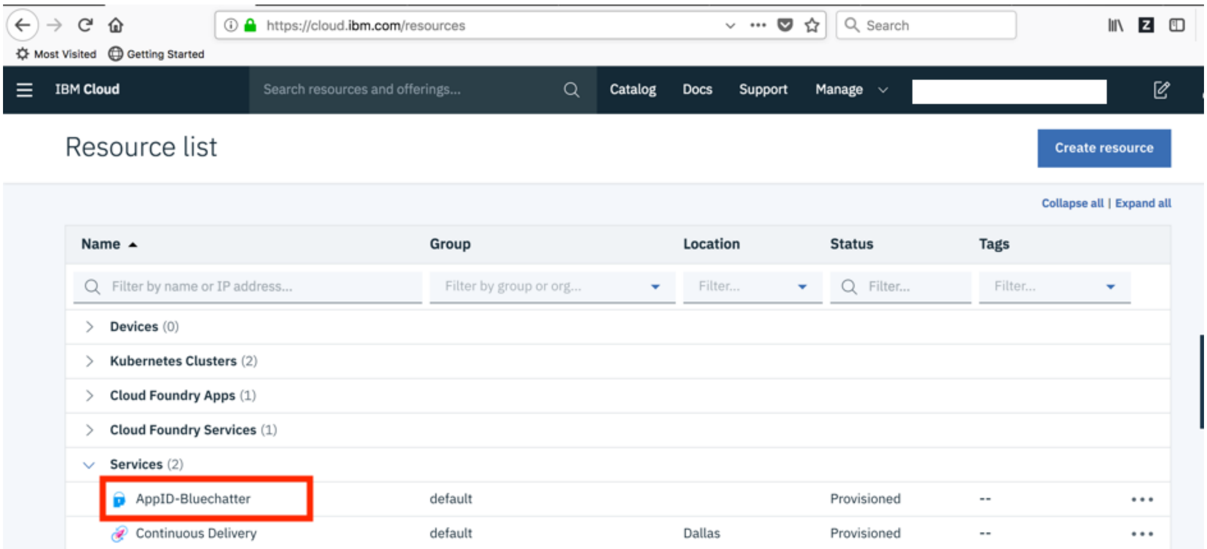
Choose a region/location to deploy in: Dallas

Select a resource group: default

Tags: Examples: env:dev, version-1

Features

You can see the App ID instance is created and listed in your resource list.

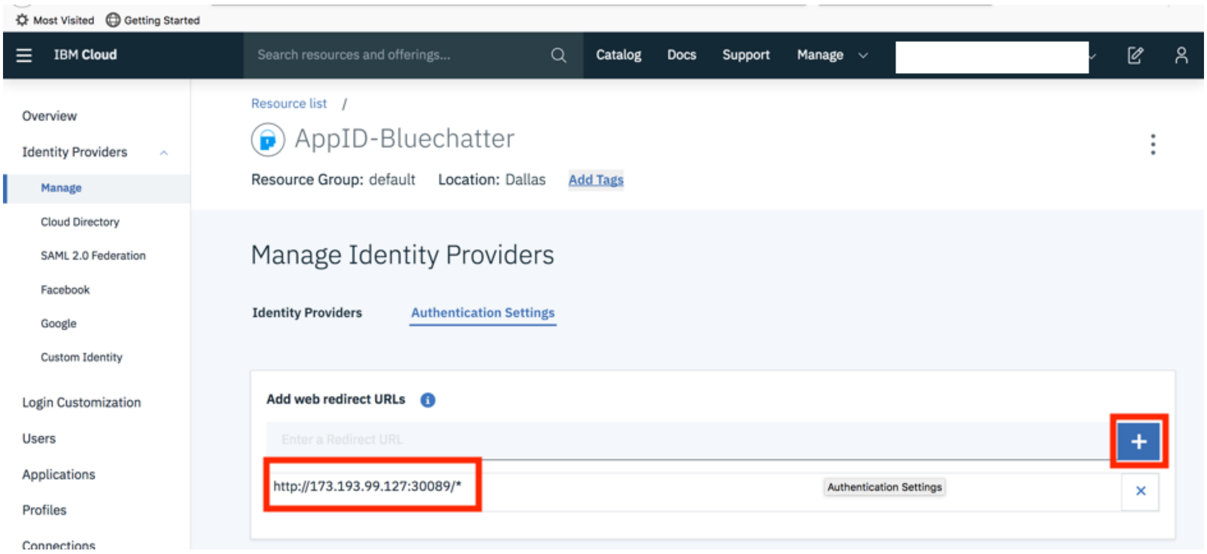

https://cloud.ibm.com/resources
Search
Most Visited
Getting Started
IBM Cloud
Search resources and offerings...
Catalog
Docs
Support
Manage
Create resource
Collapse all | Expand all

Resource list

Name ▲	Group	Location	Status	Tags	
Filter by name or IP address...	Filter by group or org...	Filter...	Filter...	Filter...	
Devices (0)					
Kubernetes Clusters (2)					
Cloud Foundry Apps (1)					
Cloud Foundry Services (1)					
Services (2)					
AppID-Bluechatter	default		Provisioned	--	•••
Continuous Delivery	default	Dallas	Provisioned	--	•••

2. Configure the Service created to use in BlueChatter Application

Click on the APP ID service instance, go to Identity Providers->Manage->Authentication Settings->Add web redirect URLs - URL shd be same as BlueChatter app link followed by /*



AppID-Bluechatter

Resource Group: default Location: Dallas

Manage Identity Providers

Add web redirect URLs

http://173.193.99.127:30089/*

Authentication Settings

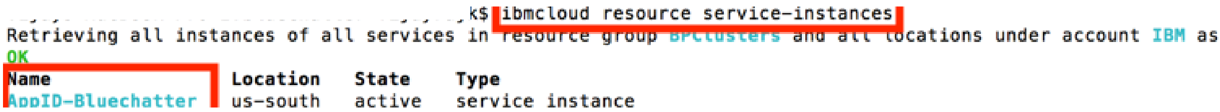
3. Bind the App ID instance to Kubernetes Cluster

You need to bind the appid instance you created with your cluster.

For doing that follow below steps

a) Get the appid-service instance by typing below command in the terminal

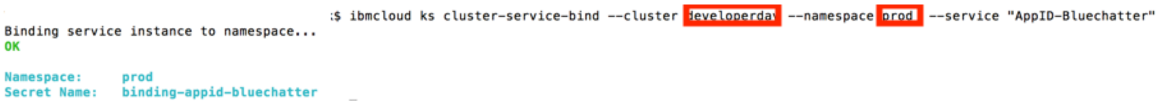
```
$ ibmcloud resource service-instances
```



```
k$ ibmcloud resource service-instances
Retrieving all instances of all services in resource group BPClusters and all locations under account IBM as
OK
Name                 Location   State    Type
AppID-Bluechatter    us-south   active   service instance
```

b) Bind the cluster with your appid instance using the below command. Make sure you replace the <cluster name> and <namespace> appropriately. This will create the required secrets.

```
     $ ibmcloud ks cluster-service-bind --cluster
<your_cluster_name> --namespace <your_namespace> --service
"AppID-Bluechatter"
```



```
:$ ibmcloud ks cluster-service-bind --cluster developerday --namespace prod --service "AppID-Bluechatter"
Binding service instance to namespace...
OK

Namespace:     prod
Secret Name:   binding-appid-bluechatter
```

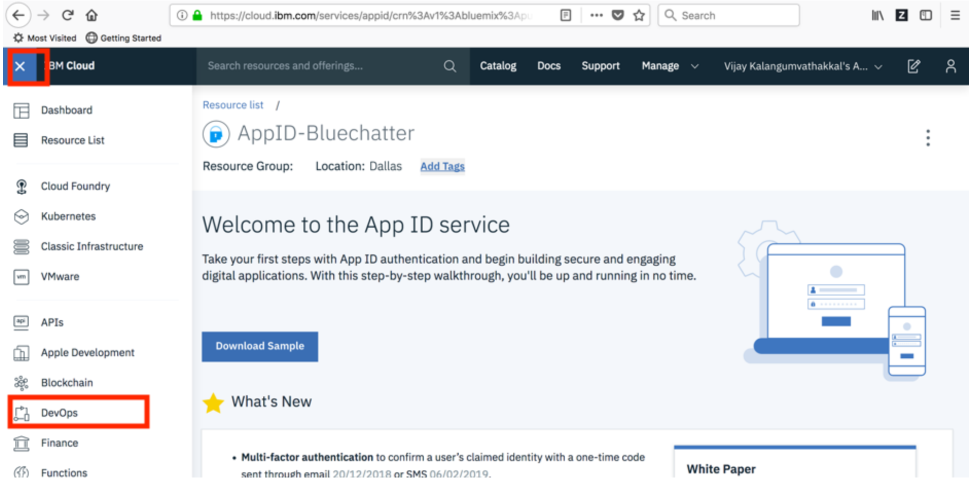
At this point the cluster has the required credentials in the form of kubernetes secrets to communicate to the App ID instance.

4. Change the application files to incorporate the Security.

Now we will deploy a new version of BlueChatter Application by making changes to the Application Code via the Tool Chain you deployed earlier.

Click on the hamburger menu from IBM Cloud portal

Select DevOps from the menu.



https://cloud.ibm.com/services/appid/crn%3Av1%3Abluemix%3Ap

AppID-Bluechatter

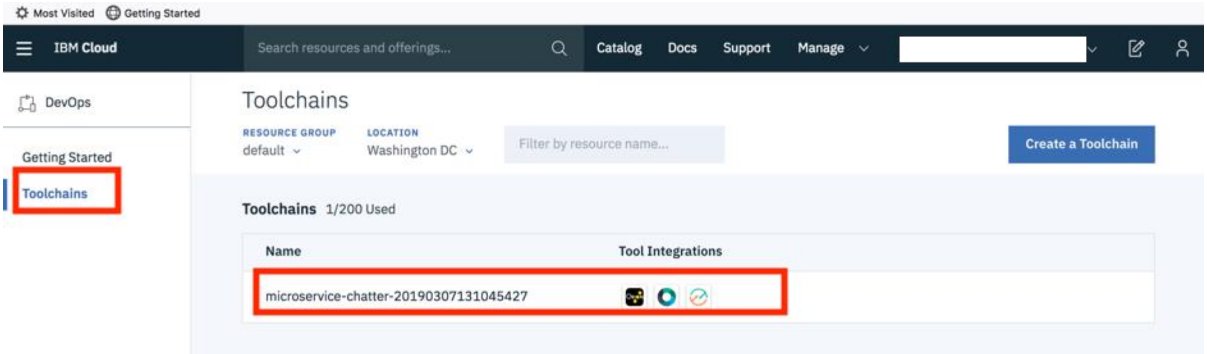
Resource Group: Location: Dallas

Welcome to the App ID service

Take your first steps with App ID authentication and begin building secure and engaging digital applications. With this step-by-step walkthrough, you'll be up and running in no time.

- **Multi-factor authentication** to confirm a user's claimed identity with a one-time code sent through email 20/12/2018 or SMS 06/02/2019.

This will open up the tool chain page. Make sure you select the appropriate location where you have created your tool chain initially.



Toolchains

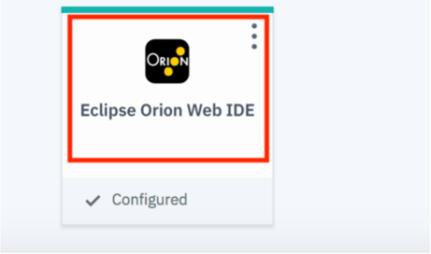
RESOURCE GROUP: default

LOCATION: Washington DC

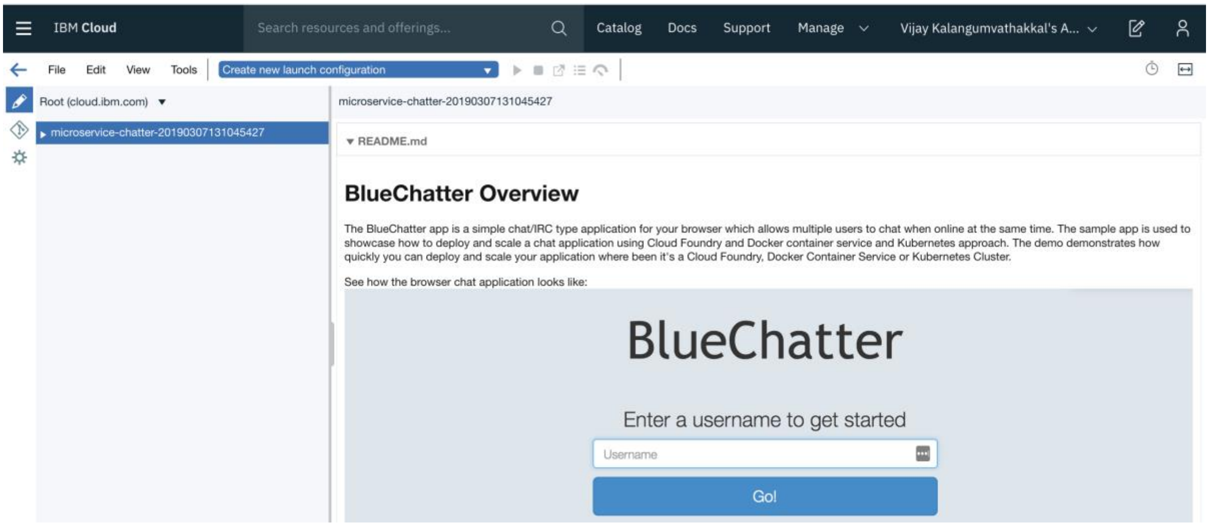
Toolchains 1/200 Used

Name	Tool Integrations
microservice-chatter-20190307131045427	

Click on “microservice-chatter” tool chain and Click on the “Eclipse Orion Web IDE”.



This will open the web base git editor. Now, You can make the code changes, required for enabling the authentication.



IBM Cloud
Search resources and offerings...
Catalog
Docs
Support
Manage
Vijay Kalangumvathakkal's A...
File
Edit
View
Tools
Create new launch configuration
Root (cloud.ibm.com)
microservice-chatter-20190307131045427
microservice-chatter-20190307131045427
README.md

BlueChatter Overview

The BlueChatter app is a simple chat/IRC type application for your browser which allows multiple users to chat when online at the same time. The sample app is used to showcase how to deploy and scale a chat application using Cloud Foundry and Docker container service and Kubernetes approach. The demo demonstrates how quickly you can deploy and scale your application where been it's a Cloud Foundry, Docker Container Service or Kubernetes Cluster.

See how the browser chat application looks like:

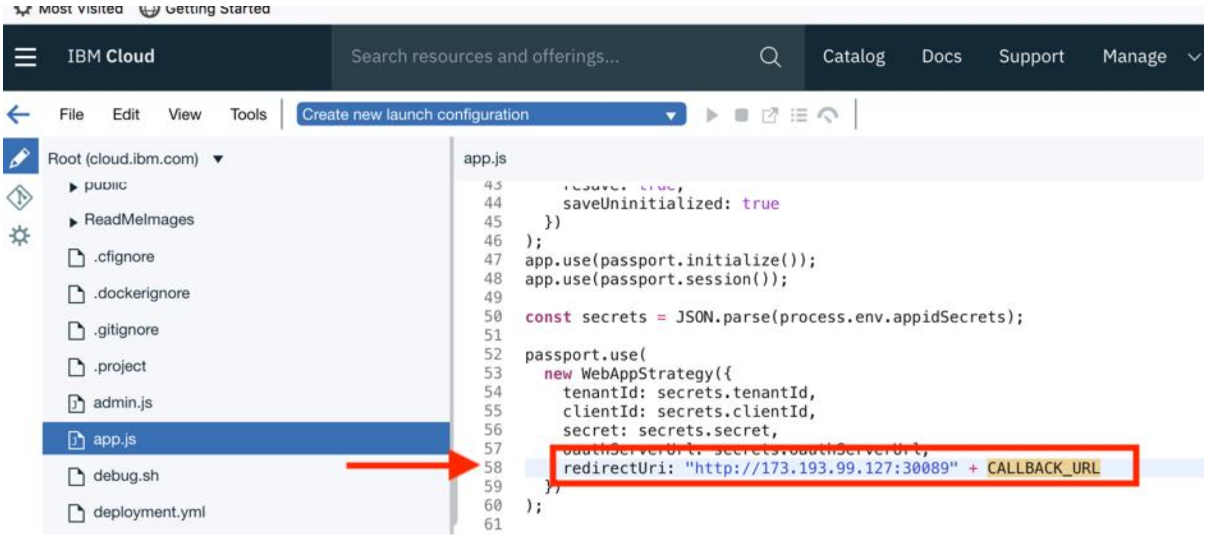
a) Now let's use in the following files from this repo - https://github.com/logeswtr/bluechatter/changedfiles and

b) Make the changes mentioned below.

You will be changing 3 files : 1) app.js 2) kubernetes.yml 3) package.json

These code changes uses the APP ID SDK and enables the authentication mechanism provided by App ID.

1) app.js –
   a. Copy the content of app.js from https://github.com/logeswtr/bluechatter/changedfiles and paste into the app.js in the tool chain editor.
      Line number 58, please make sure the URL is same BlueChatter app link.



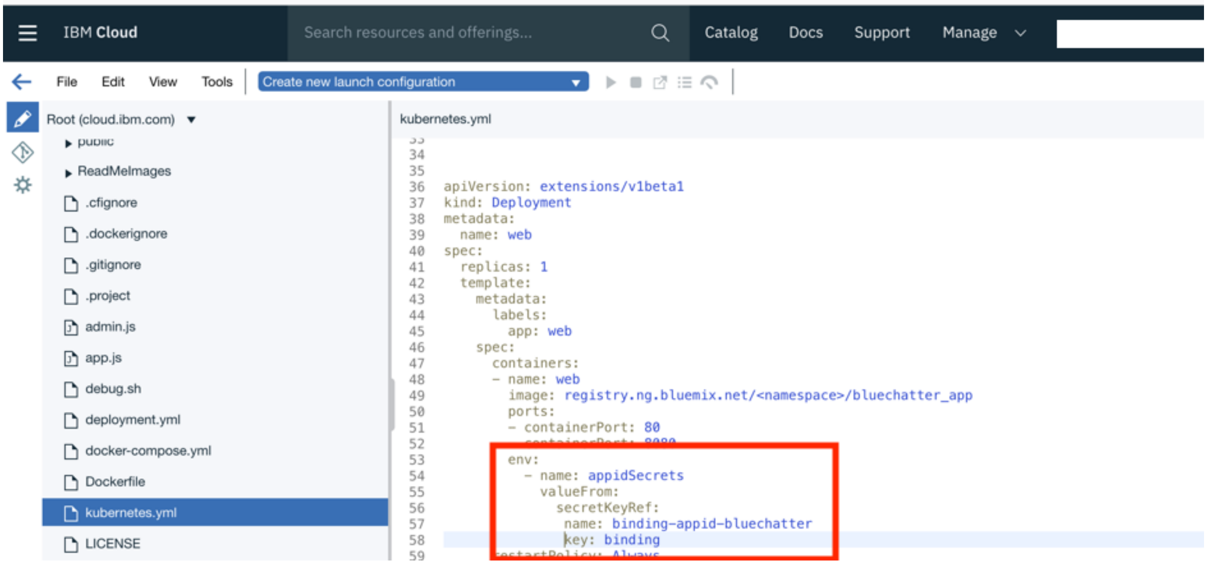
```
    resave: true,
    saveUninitialized: true
  })
);
app.use(passport.initialize());
app.use(passport.session());

const secrets = JSON.parse(process.env.appidSecrets);

passport.use(
  new WebAppStrategy({
    tenantId: secrets.tenantId,
    clientId: secrets.clientId,
    secret: secrets.secret,
    oauthServerUrl: secrets.oauthServerUrl,
    redirectUri: "http://173.193.99.127:30089" + CALLBACK_URL
  })
);
```

Save the file in the editor.

2) kubernetes.yml -

   a. Copy **only env** related lines alone from kubernetes.yml file from https://github.com/logeswtr/bluechatter/changedfiles and paste into the kubernetes.yml in the tool chain editor under the **deployment** named web, Please refer screenshot below .

```
env:
   - name: appidSecrets
     valueFrom:
       secretKeyRef:
         name: binding-appid-bluechatter
          key: binding
```



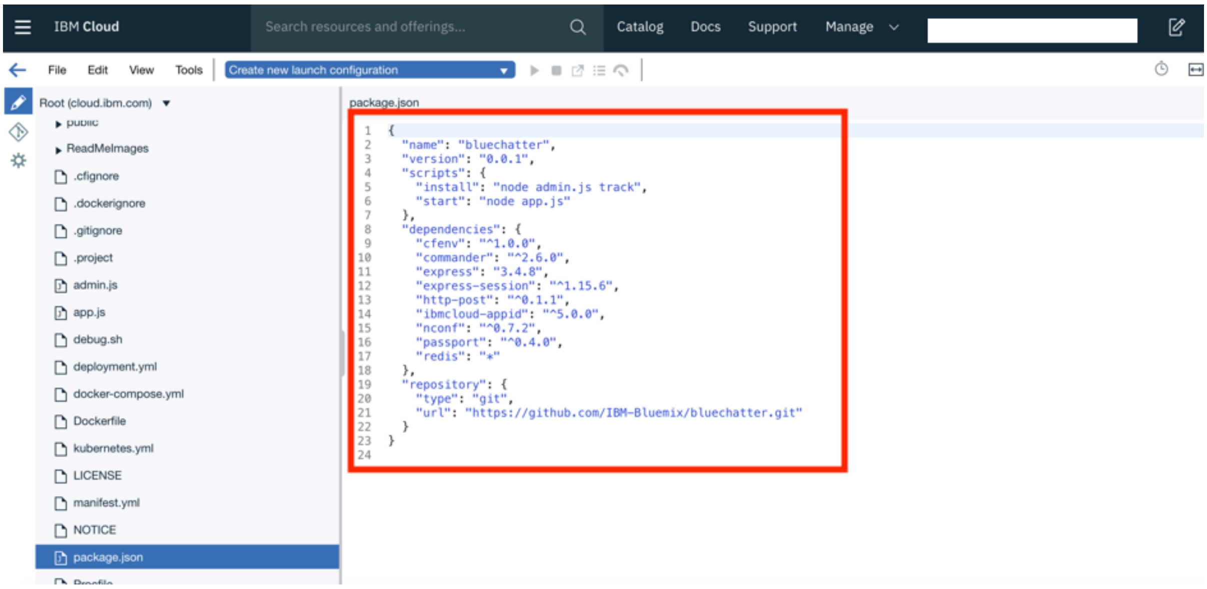
```yaml
apiVersion: extensions/v1beta1
kind: Deployment
metadata:
  name: web
spec:
  replicas: 1
  template:
    metadata:
      labels:
        app: web
    spec:
      containers:
      - name: web
        image: registry.ng.bluemix.net/<namespace>/bluechatter_app
        ports:
        - containerPort: 80
        - containerPort: 8080
        env:
          - name: appidSecrets
            valueFrom:
              secretKeyRef:
               name: binding-appid-bluechatter
               key: binding
      restartPolicy: Always
```

Save the file in the editor

3) package.json –

   a. Copy the full content of package.json from https://github.com/logeswtr/bluechatter/changedfiles and paste into the package.json in the tool chain editor

```
{

"name": "bluechatter",

"version": "0.0.1",

"scripts": {

"install": "node admin.js track",

"start": "node app.js"

},

"dependencies": {

"cfenv": "^1.0.0",

"commander": "^2.6.0",

"express": "3.4.8",

"express-session": "^1.15.6",

"http-post": "^0.1.1",
"ibmcloud-appid": "^5.0.0",

"nconf": "^0.7.2",

"passport": "^0.4.0",

"redis": "*"

},

"repository": {

"type": "git",

"url": "https://github.com/IBM-Bluemix/bluechatter.git"

}

}
```



```
{
  "name": "bluechatter",
  "version": "0.0.1",
  "scripts": {
    "install": "node admin.js track",
    "start": "node app.js"
  },
  "dependencies": {
    "cfenv": "^1.0.0",
    "commander": "^2.6.0",
    "express": "3.4.8",
    "express-session": "^1.15.6",
    "http-post": "^0.1.1",
    "ibmcloud-appid": "^5.0.0",
    "nconf": "^0.7.2",
    "passport": "^0.4.0",
    "redis": "*"
  },
  "repository": {
    "type": "git",
    "url": "https://github.com/IBM-Bluemix/bluechatter.git"
  }
}
```

Save the file in the editor.

This complete the task of Application Code changes for adding security

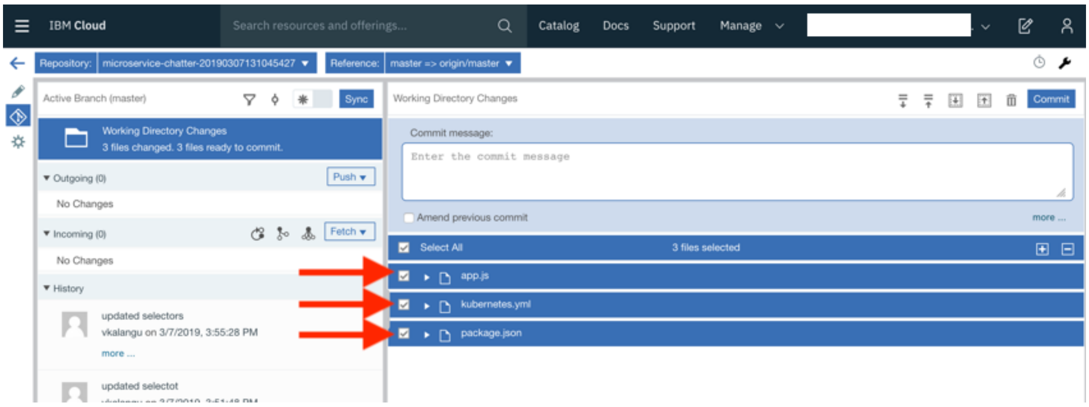
Task 3 Push the changes, Trigger new build and deployment through ToolChain

Overview

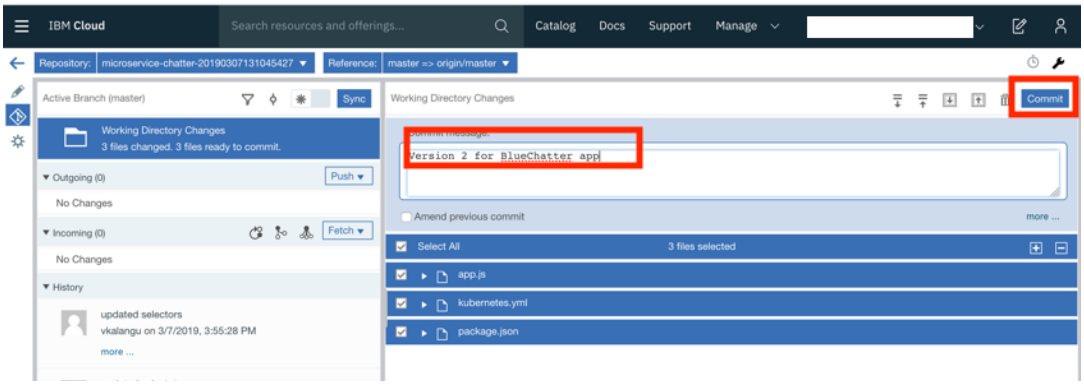
This new version of BlueChatter Application changes is saved through editor in repository. Now you will push the committed code changes by clicking the push button. This will trigger a new build in the tool chain. And after successful build you will be asked to use this authentication method to access the application.

1. Verify the application files and Push changes from GIT.

a) Click on the git view. You can see the 3 file changes displayed.

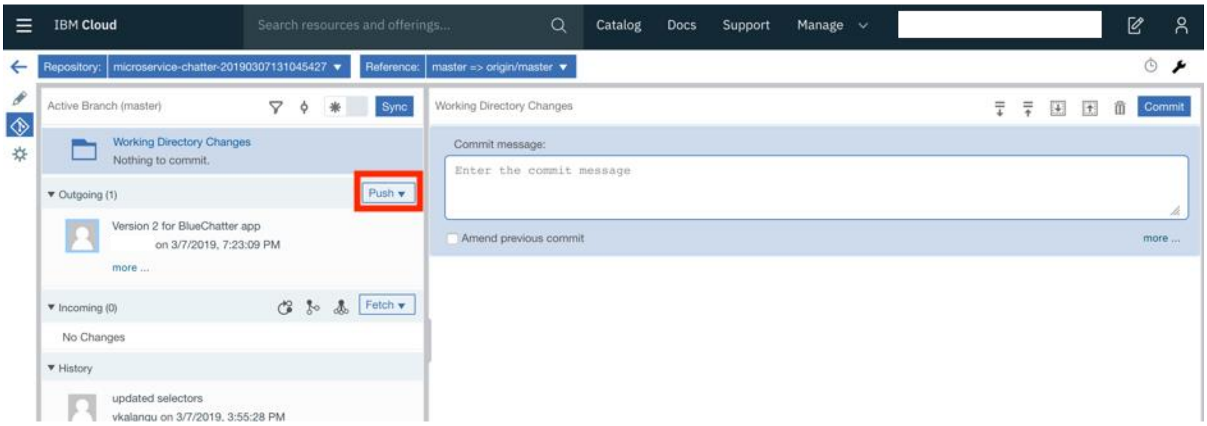


b) Commit the changes by clicking the commit button and providing the appropriate commit messages.



Commit message: version 2 for BlueChatter app

c) Push the committed code changes by clicking the push button.

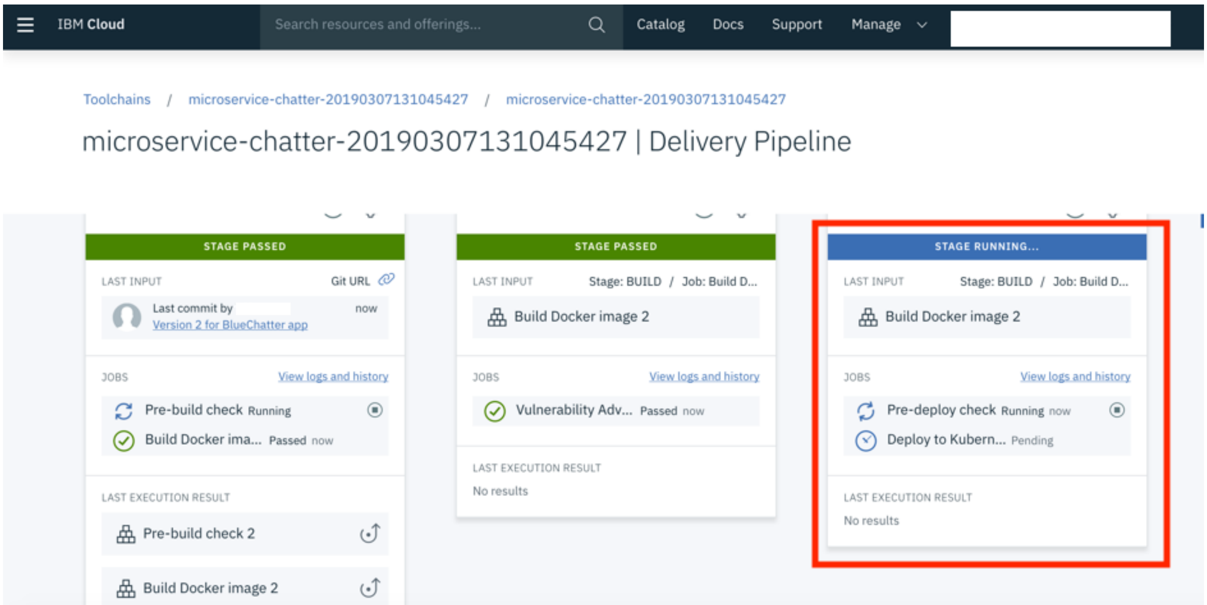


Version 2 for BlueChatter app
on 3/7/2019, 7:23:09 PM

updated selectors
vkalangu on 3/7/2019, 3:55:28 PM

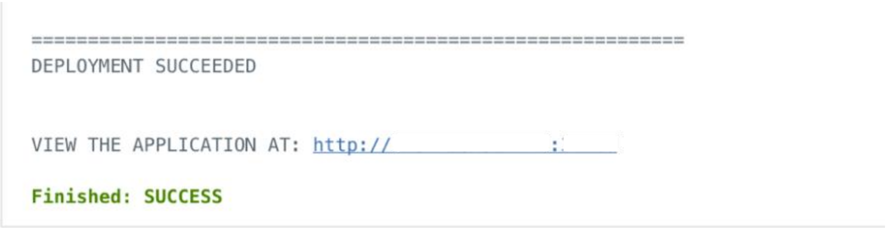
2. Push changes from GIT, triggers ToolChain for new build.

This will trigger a new build in the tool chain. You can see this by going back to your tool chain as below.


IBM Cloud
Search resources and offerings...
Catalog
Docs
Support
Manage
Toolchains / microservice-chatter-20190307131045427 / microservice-chatter-20190307131045427
microservice-chatter-20190307131045427 | Delivery Pipeline
STAGE PASSED
LAST INPUT
Git URL
Last commit by
now
Version 2 for BlueChatter app
JOBS
View logs and history
Pre-build check Running
Build Docker ima... Passed now
LAST EXECUTION RESULT
Pre-build check 2
Build Docker image 2
STAGE PASSED
LAST INPUT
Stage: BUILD / Job: Build D...
Build Docker image 2
JOBS
View logs and history
Vulnerability Adv... Passed now
LAST EXECUTION RESULT
No results
STAGE RUNNING...
LAST INPUT
Stage: BUILD / Job: Build D...
Build Docker image 2
JOBS
View logs and history
Pre-deploy check Running now
Deploy to Kubern... Pending
LAST EXECUTION RESULT
No results

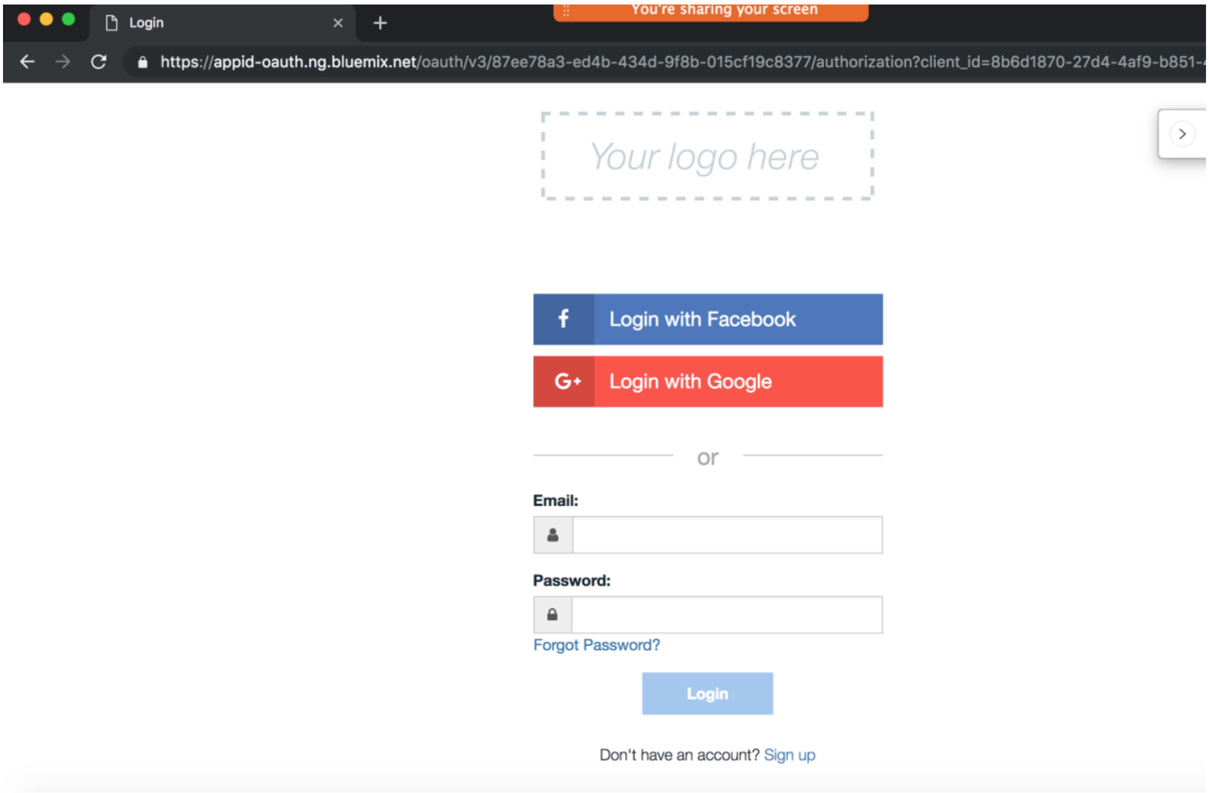
3. Build successful and New version of Application is available.

Eventually all the stages will be in passed stated. Now you can hit the BlueChatter App URL in the browser.



```
==========================================================
DEPLOYMENT SUCCEEDED


VIEW THE APPLICATION AT: http://                    :

Finished: SUCCESS
```

You will be asked for authentication before you can access BlueChatter Application



https://appid-oauth.ng.bluemix.net/oauth/v3/87ee78a3-ed4b-434d-9f8b-015cf19c8377/authorization?client_id=8b6d1870-27d4-4af9-b851-

Email:

Password:

Forgot Password?

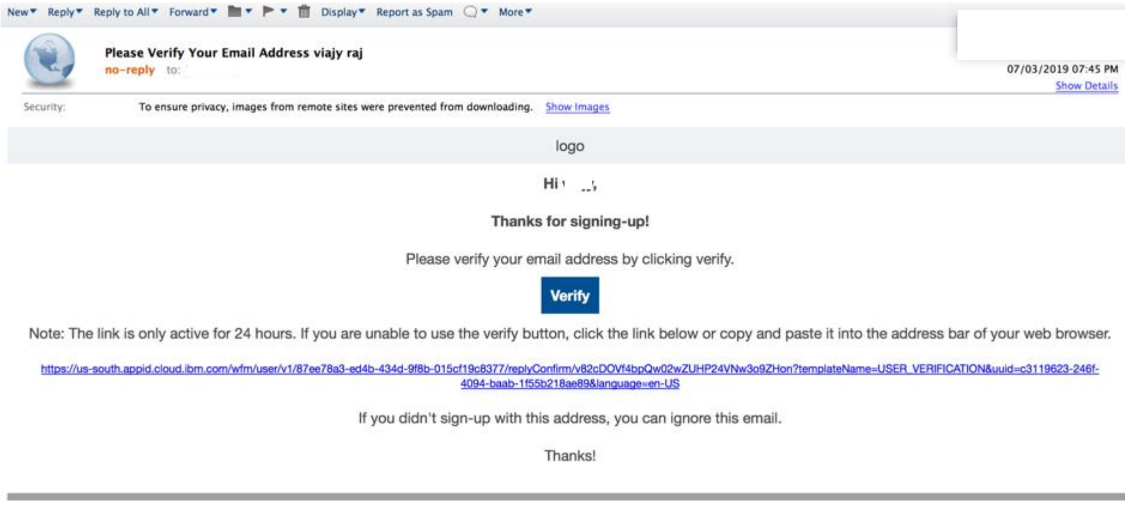
Don't have an account? Sign up

You have.3 choices , sign in with

- Facebook
- Google
- You can sign up using the Sign up link.

4. Signup and Access Application.

You will get a confirmation mail to the mail is given during the sign up.



Please Verify Your Email Address viajy raj

no-reply to:

07/03/2019 07:45 PM

Show Details

Security: To ensure privacy, images from remote sites were prevented from downloading. Show Images

Hi

Thanks for signing-up!

Please verify your email address by clicking verify.

Note: The link is only active for 24 hours. If you are unable to use the verify button, click the link below or copy and paste it into the address bar of your web browser.

https://us-south.appid.cloud.ibm.com/wfm/user/v1/87ee78a3-ed4b-434d-9f8b-015cf19c8377/replyConfirm/v82cDOVf4bpQw02wZUHP24VNw3o9ZHon?templateName=USER_VERIFICATION&uuid=c3119623-246f-4094-baab-1f55b218ae89&language=en-US

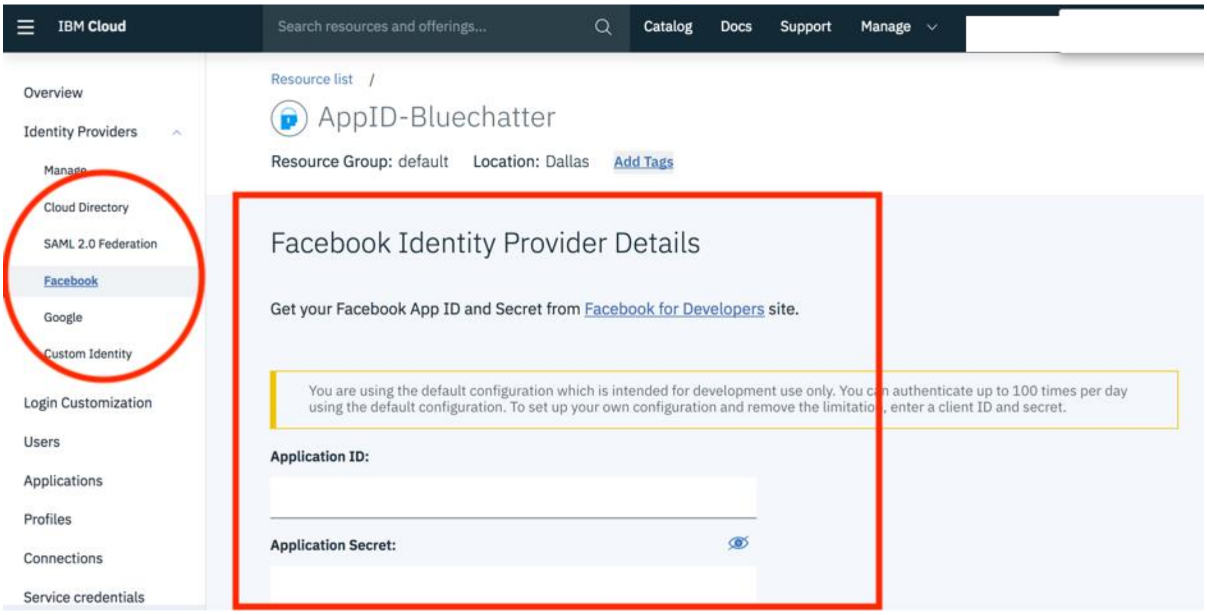
If you didn't sign-up with this address, you can ignore this email.

Thanks!

a) Verify the email address and continue accessing the BlueChatter app
b) After authenticating with the userid/password you created.

5. Use Facebook/Google Identity Providers to Access Application.

You can also use the Facebook/Google Identity Providers too after you follow the configuration mentioned in the App ID



AppID-Bluechatter

Resource Group: default   Location: Dallas   Add Tags

Facebook Identity Provider Details

Get your Facebook App ID and Secret from Facebook for Developers site.

You are using the default configuration which is intended for development use only. You can authenticate up to 100 times per day using the default configuration. To set up your own configuration and remove the limitation, enter a client ID and secret.

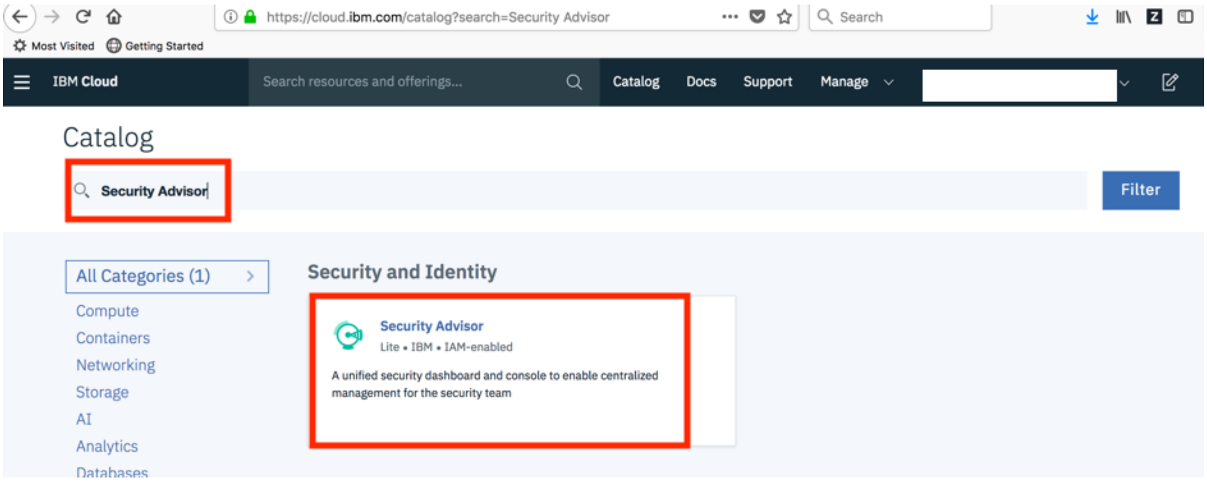
Application ID:

Application Secret:

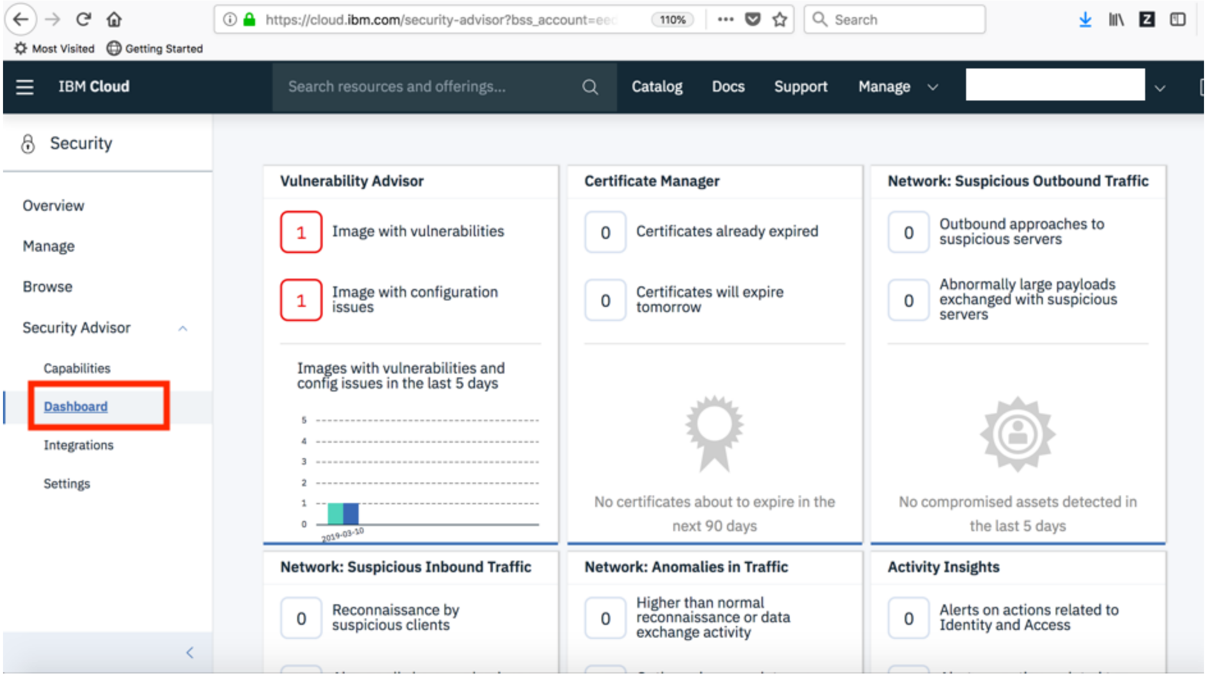
Task 4 Gain visibility into any vulnerabilities via Security Advisor dashboard.

IBM Cloud Security Advisor is a security dashboard that provides centralized security management. The dashboard unifies vulnerability and network data as well as application and system findings from IBM Services, partners and user-defined sources. By centralizing visibility and enabling drill down to resolution, Security Advisor empowers the security admin to cohesively manage security on IBM Cloud workloads.

Open the Security Advisor Dashboard from the Catalog.



You will be able to see security posture of various components in your app.



Vulnerability Advisor

1 Image with vulnerabilities

1 Image with configuration issues

Images with vulnerabilities and config issues in the last 5 days

Certificate Manager

0 Certificates already expired

0 Certificates will expire tomorrow

No certificates about to expire in the next 90 days

Network: Suspicious Outbound Traffic

0 Outbound approaches to suspicious servers

0 Abnormally large payloads exchanged with suspicious servers

No compromised assets detected in the last 5 days

Network: Suspicious Inbound Traffic

0 Reconnaissance by suspicious clients

Network: Anomalies in Traffic

0 Higher than normal reconnaissance or data exchange activity

Activity Insights

0 Alerts on actions related to Identity and Access

Congratulations! You have completed the lab! In case you have more time, you can do optional tasks

Task 5 Debug and log your Kubernetes applications

(Optional)

This is an optional task, To understand more about the Kubernetes cluster and how to debug and get logs from your application.

Use instruction from below link
https://developer.ibm.com/tutorials/debug-and-log-your-kubernetes-app/

References

IBM Kubernetes Service - https://cloud.ibm.com/docs/containers?topic=containers-container_index#container_index

IBM Kubernetes Service Solution tutorials - https://cloud.ibm.com/docs/tutorials?topic=solution-tutorials-scalable-webapp-kubernetes#scalable-webapp-kubernetes

IBM Cloud DevOps Insights - https://cloud.ibm.com/docs/services/DevOpsInsights?topic=DevOpsInsights-getting-started#getting-started

App ID - https://www.ibm.com/blogs/bluemix/2018/04/ibm-cloud-app-id-technical-white-paper-now-available/

IBM Cloud Security Advisor - https://cloud.ibm.com/docs/services/security-advisor?topic=security-advisor-index#index

Debug and log your Kubernetes applications
https://developer.ibm.com/tutorials/debug-and-log-your-kubernetes-app/