Zero to Cloud-Native with IBM Cloud

Part 7: Preparing to Deploy the Application

Kevin Collins
kevincollins@us.ibm.com
Technical Sales Leader
IBM Cloud Enterprise Containers – Americas

Kunal Malhotra
kunal.malhotra3@ibm.com
Cloud Platform Engineer
IBM Cloud MEA

1 - Preparing to Deploy the Application

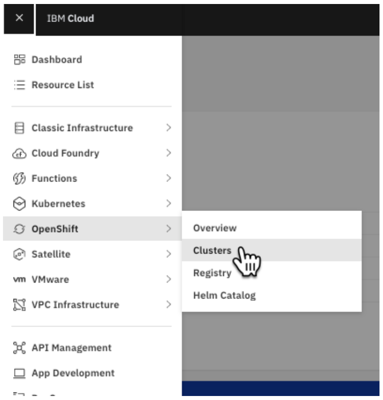
Now that we have provisioned all the IBM Cloud Services our application will use and we have a solid development environment setup, we can now configure all the services so we can deploy our tutorial application.

1 -1 RedHat OpenShift on IBM Cloud

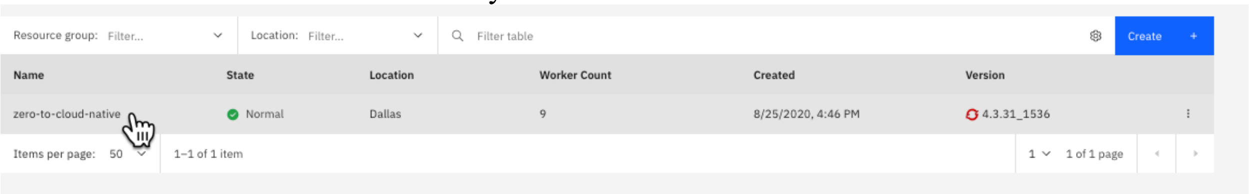
Starting with RedHat OpenShift on IBM Cloud, there are a couple of things we need to do to prepare our cluster to deploy our application.

To begin with, we are going to install IBM Log Analysis with LogDNA and IBM Monitoring with SysDig.   Future sections will go through a deep dive on using both LogDNA and SysDig. This section will just go through the install process so logs and metric data will start flowing to LogDNA and SysDig.

To get started, we are going to navigate to our OpenShift cluster. From the IBM Cloud Console, click on the hamburger menu -> OpenShift -> clusters.



The next screen will show a list of all your clusters. Click on the zero-to-cloud-native cluster.



Resource group: Filter... Location: Filter... Filter table

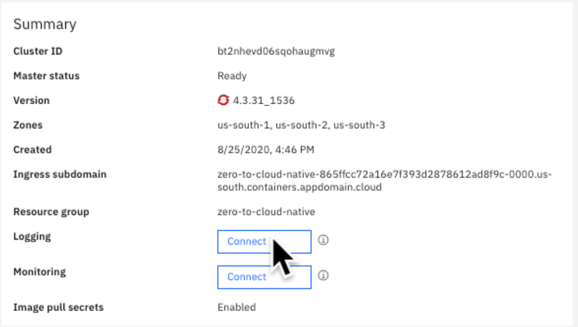
Name	State	Location	Worker Count	Created	Version
zero-to-cloud-native	Normal	Dallas	9	8/25/2020, 4:46 PM	4.3.31_1536

Items per page: 50 1–1 of 1 item 1 1 of 1 page

The next screen will show an Overview of your cluster. You can high level information about the status of your cluster, CPU and memory usage.

2 Configure Logging

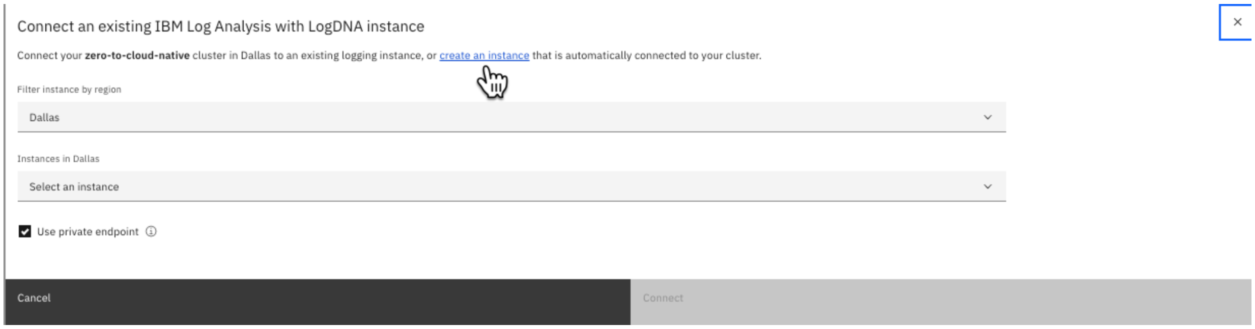
We will start by configuring logging for our cluster. IBM Cloud OpenShift has made it incredibly easy to enable Logging and Monitoring with tight integration to both services. Starting with enabling logging, click on ‘Connect’ next to Logging.



Summary

Cluster ID	bt2nhevd06sqohaugmvg
Master status	Ready
Version	4.3.31_1536
Zones	us-south-1, us-south-2, us-south-3
Created	8/25/2020, 4:46 PM
Ingress subdomain	zero-to-cloud-native-865ffcc72a16e7f393d2878612ad8f9c-0000.us-south.containers.appdomain.cloud
Resource group	zero-to-cloud-native
Logging	Connect
Monitoring	Connect
Image pull secrets	Enabled

On the next screen you will have an option to either create new instance of IBM Log Analysis with LogDNA or create a new instance. We will be creating a new instance, click on create an instance.



Connect an existing IBM Log Analysis with LogDNA instance

Connect your **zero-to-cloud-native** cluster in Dallas to an existing logging instance, or create an instance that is automatically connected to your cluster.

Filter instance by region: Dallas

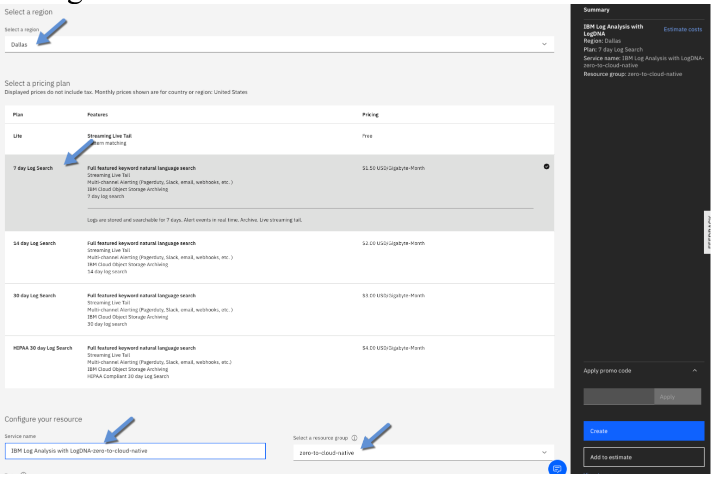
Instances in Dallas: Select an instance

☑ Use private endpoint

On the next screen, enter Dallas or the same region you have been using throughout the tutorial. Select 7 day Log Search as we will be going through more advanced features of LogDNA such as integrating alerts with Slack in future tutorials.

Give the service a meaningful name, I will use IBM Log Analysis with LogDNA-zero-to-cloud-native.

Finally, make sure to select the zero-to-cloud-native resource group we have been using throughout the tutorial.



Select a region

Select a region: Dallas

Select a pricing plan

Displayed prices do not include tax. Monthly prices shown are for country or region: United States

Plan	Features	Pricing
Lite	**Streaming Live Tail** Pattern matching	Free
7 day Log Search	**Full featured keyword natural language search** Streaming Live Tail Multi-channel Alerting (Pagerduty, Slack, email, webhooks, etc.) IBM Cloud Object Storage Archiving 7 day log search Logs are stored and searchable for 7 days. Alert events in real time. Archive. Live streaming tail.	$1.50 USD/Gigabyte-Month
14 day Log Search	**Full featured keyword natural language search** Streaming Live Tail Multi-channel Alerting (Pagerduty, Slack, email, webhooks, etc.) IBM Cloud Object Storage Archiving 14 day log search	$2.00 USD/Gigabyte-Month
30 day Log Search	**Full featured keyword natural language search** Streaming Live Tail Multi-channel Alerting (Pagerduty, Slack, email, webhooks, etc.) IBM Cloud Object Storage Archiving 30 day log search	$3.00 USD/Gigabyte-Month
HIPAA 30 day Log Search	**Full featured keyword natural language search** Streaming Live Tail Multi-channel Alerting (Pagerduty, Slack, email, webhooks, etc.) IBM Cloud Object Storage Archiving HIPAA Compliant 30 day Log Search	$4.00 USD/Gigabyte-Month

Configure your resource

Service name: IBM Log Analysis with LogDNA-zero-to-cloud-native

Select a resource group: zero-to-cloud-native

Summary

IBM Log Analysis with LogDNA

Region: Dallas

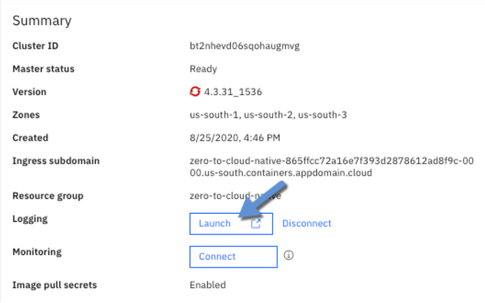
Plan: 7 day Log Search

Service name: IBM Log Analysis with LogDNA-zero-to-cloud-native

Resource group: zero-to-cloud-native

After entering all these settings click create.

You will now see on the cluster overview screen that you can launch the Logging dashboard. Future tutorials will go through how to use IBM Log Analysis with LogDNA.



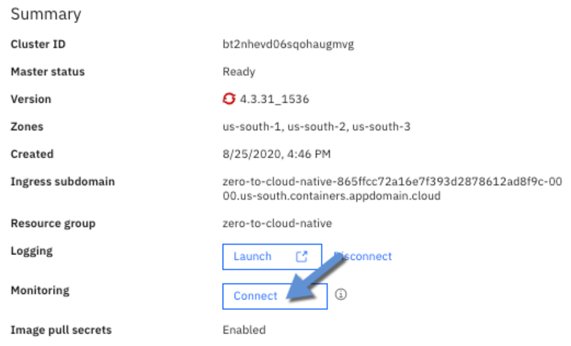
Summary

Cluster ID	bt2nhevd06sqohaugmvg
Master status	Ready
Version	4.3.31_1536
Zones	us-south-1, us-south-2, us-south-3
Created	8/25/2020, 4:46 PM
Ingress subdomain	zero-to-cloud-native-865ffcc72a16e7f393d2878612ad8f9c-0000.us-south.containers.appdomain.cloud
Resource group	zero-to-cloud-native
Logging	Launch Disconnect
Monitoring	Connect
Image pull secrets	Enabled

3 - Configure Monitoring

Next, we will follow a similar process to connect the IBM Cloud Monitoring with SysDig service.

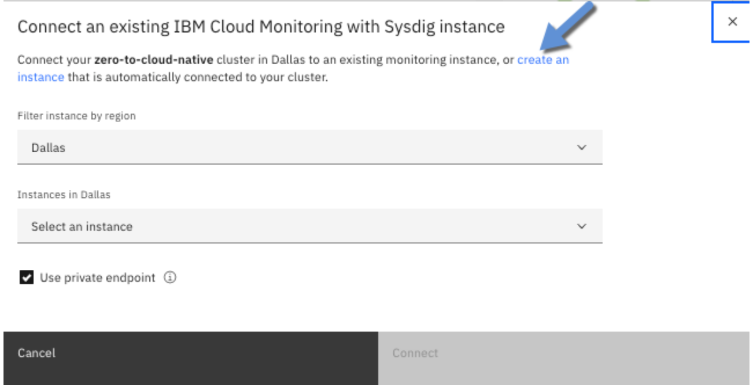
Click the connect button next to Monitoring.



Summary

Cluster ID	bt2nhevd06sqohaugmvg
Master status	Ready
Version	4.3.31_1536
Zones	us-south-1, us-south-2, us-south-3
Created	8/25/2020, 4:46 PM
Ingress subdomain	zero-to-cloud-native-865ffcc72a16e7f393d2878612ad8f9c-0000.us-south.containers.appdomain.cloud
Resource group	zero-to-cloud-native
Logging	Launch Disconnect
Monitoring	Connect
Image pull secrets	Enabled

As with LogDNA we can either connect SysDig to an existing instance or create a new instance. In this tutorial we will be creating a new instance, click on create an instance.



Connect an existing IBM Cloud Monitoring with Sysdig instance

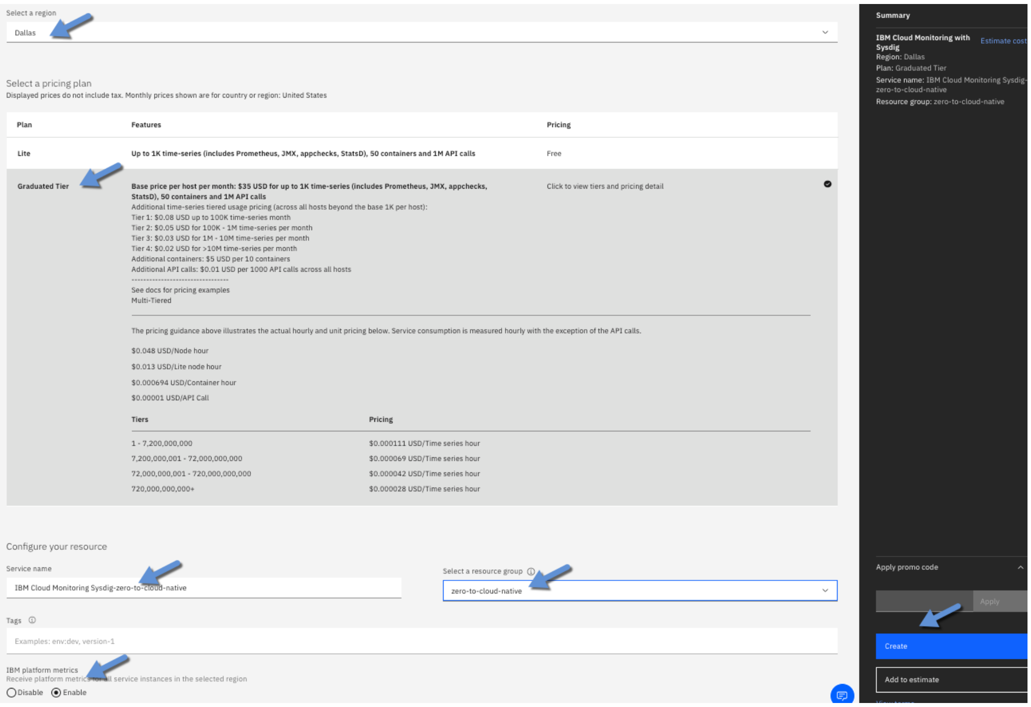
Connect your **zero-to-cloud-native** cluster in Dallas to an existing monitoring instance, or create an instance that is automatically connected to your cluster.

Filter instance by region: Dallas

Instances in Dallas: Select an instance

☑ Use private endpoint

On the next screen, enter Dallas or the same region you have been using throughout the tutorial.
Select Graduated Tier as the tutorial uses advance features of SysDig.
Give the service a meaningful name, I will use IBM Cloud Monitoring Sysdig-zero-to-cloud-native
Make sure to select the zero-to-cloud-native resource group we have been using throughout the tutorial.
Finally, enable IBM platform metrics so we can have a single dashboard which will also monitor all the cloud services we have deployed.



Select a region

Dallas

Select a pricing plan

Displayed prices do not include tax. Monthly prices shown are for country or region: United States

Plan	Features	Pricing
Lite	**Up to 1K time-series (includes Prometheus, JMX, appchecks, StatsD), 50 containers and 1M API calls**	Free
Graduated Tier	**Base price per host per month: $35 USD for up to 1K time-series (includes Prometheus, JMX, appchecks, StatsD), 50 containers and 1M API calls** Additional time-series tiered usage pricing (across all hosts beyond the base 1K per host): Tier 1: $0.08 USD up to 100K time-series month Tier 2: $0.05 USD for 100K - 1M time-series per month Tier 3: $0.03 USD for 1M - 10M time-series per month Tier 4: $0.02 USD for >10M time-series per month Additional containers: $5 USD per 10 containers Additional API calls: $0.01 USD per 1000 API calls across all hosts -------------------------------- See docs for pricing examples Multi-Tiered	Click to view tiers and pricing detail

The pricing guidance above illustrates the actual hourly and unit pricing below. Service consumption is measured hourly with the exception of the API calls.

$0.048 USD/Node hour

$0.013 USD/Lite node hour

$0.000694 USD/Container hour

$0.00001 USD/API Call

Tiers	Pricing
1 - 7,200,000,000	$0.000111 USD/Time series hour
7,200,000,001 - 72,000,000,000	$0.000069 USD/Time series hour
72,000,000,001 - 720,000,000,000	$0.000042 USD/Time series hour
720,000,000,000+	$0.000028 USD/Time series hour

Configure your resource

Service name: IBM Cloud Monitoring Sysdig-zero-to-cloud-native

Select a resource group: zero-to-cloud-native

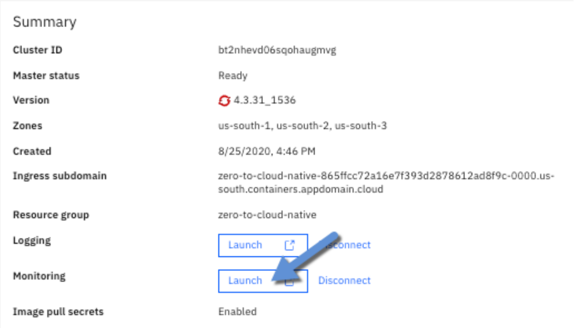
Tags: Examples: env:dev, version-1

IBM platform metrics
Receive platform metrics for all service instances in the selected region
○ Disable ● Enable

Summary

IBM Cloud Monitoring with Sysdig
Region: Dallas
Plan: Graduated Tier
Service name: IBM Cloud Monitoring Sysdig-zero-to-cloud-native
Resource group: zero-to-cloud-native

After entering all these settings, click Create. As we saw with Logging, the Monitoring line will now have an option to Launch the monitoring dashboard. A future tutorial will go through a deep dive on using SysDig to monitor our application.



Summary

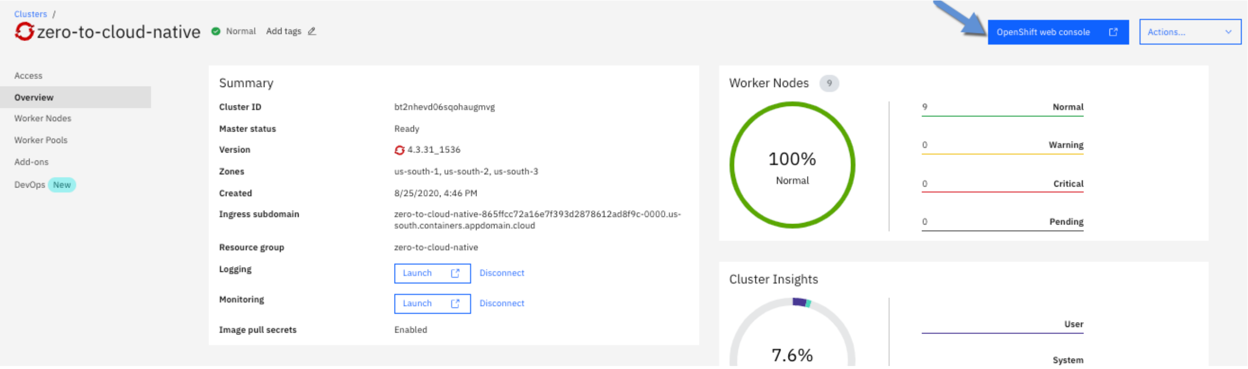
Cluster ID	bt2nhevd06sqohaugmvg
Master status	Ready
Version	4.3.31_1536
Zones	us-south-1, us-south-2, us-south-3
Created	8/25/2020, 4:46 PM
Ingress subdomain	zero-to-cloud-native-865ffcc72a16e7f393d2878612ad8f9c-0000.us-south.containers.appdomain.cloud
Resource group	zero-to-cloud-native
Logging	Launch Disconnect
Monitoring	Launch Disconnect
Image pull secrets	Enabled

4 - OpenShift

4 – 1 Create an OpenShift Project

Next, we need to create a namespace or OpenShift project where our application will run. To create a new OpenShift project, we need to first log into our OpenShift cluster and retrieve a login token.

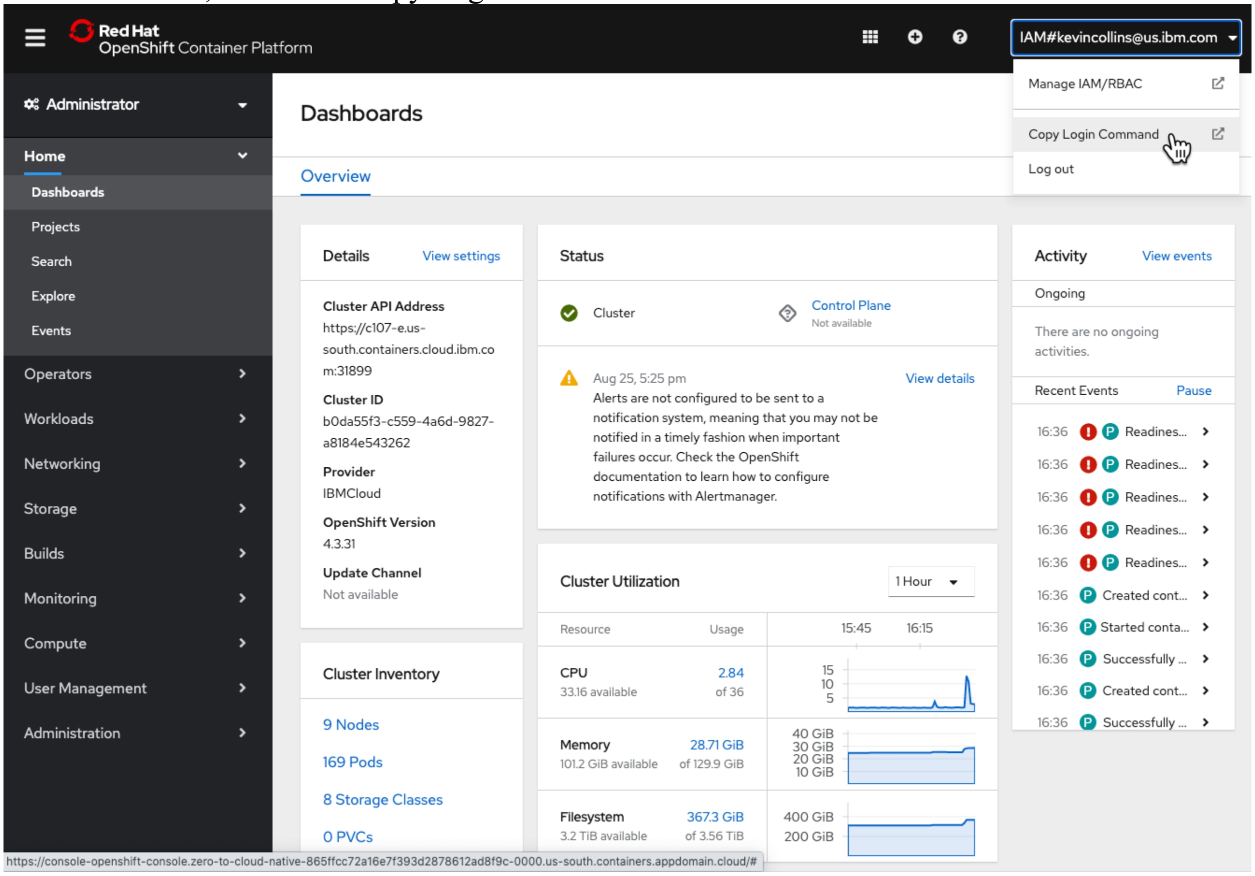
From the detailed view of your cluster, click on OpenShift web console.



Summary

Cluster ID	bt2nhevd06sqohaugmvg
Master status	Ready
Version	4.3.31_1536
Zones	us-south-1, us-south-2, us-south-3
Created	8/25/2020, 4:46 PM
Ingress subdomain	zero-to-cloud-native-865ffcc72a16e7f393d2878612ad8f9c-0000.us-south.containers.appdomain.cloud
Resource group	zero-to-cloud-native
Logging	Launch Disconnect
Monitoring	Launch Disconnect
Image pull secrets	Enabled

This will launch the OpenShift web console. To retrieve a login token, click on you IBM Cloud IAM user name, and select Copy Login Command.



Cluster API Address
https://c107-e.us-south.containers.cloud.ibm.com:31899

Cluster ID
b0da55f3-c559-4a6d-9827-a8184e543262

Provider
IBMCloud

OpenShift Version
4.3.31

Update Channel
Not available

Aug 25, 5:25 pm
Alerts are not configured to be sent to a notification system, meaning that you may not be notified in a timely fashion when important failures occur. Check the OpenShift documentation to learn how to configure notifications with Alertmanager.

There are no ongoing activities.

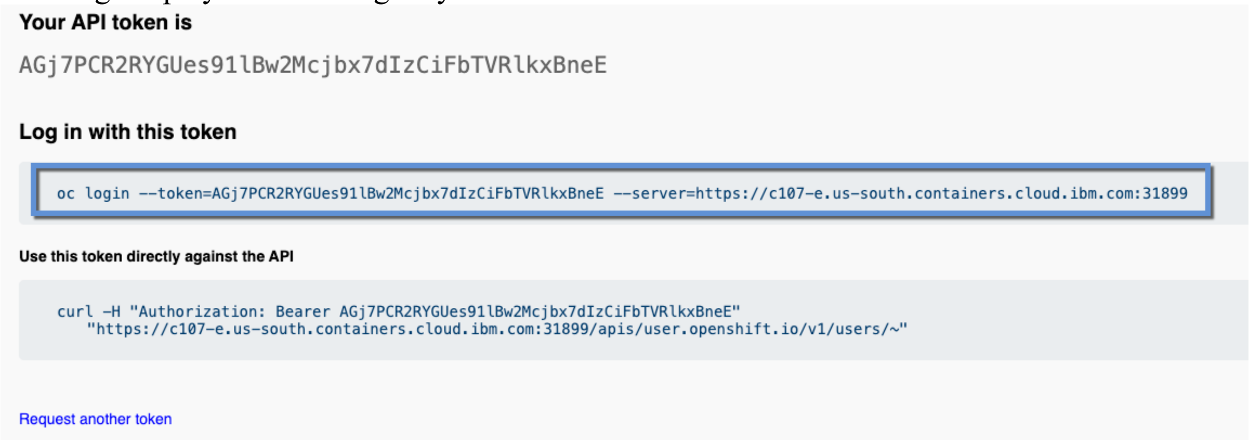
Resource	Usage	
CPU 33.16 available	2.84 of 36	
Memory 101.2 GiB available	28.71 GiB of 129.9 GiB	
Filesystem 3.2 TiB available	367.3 GiB of 3.56 TiB	

```
16:36 ! P Readines...
16:36 ! P Readines...
16:36 ! P Readines...
16:36 ! P Readines...
16:36 ! P Readines...
16:36 P Created cont...
16:36 P Started conta...
16:36 P Successfully ...
16:36 P Created cont...
16:36 P Successfully ...
```

https://console-openshift-console.zero-to-cloud-native-865ffcc72a16e7f393d2878612ad8f9c-0000.us-south.containers.appdomain.cloud/#

This will bring you to a page where you need to click Display Token.

Clicking Display Token will give you a screen like the one below.



Your API token is

AGj7PCR2RYGUes91lBw2Mcjbx7dIzCiFbTVRlkxBneE

Log in with this token

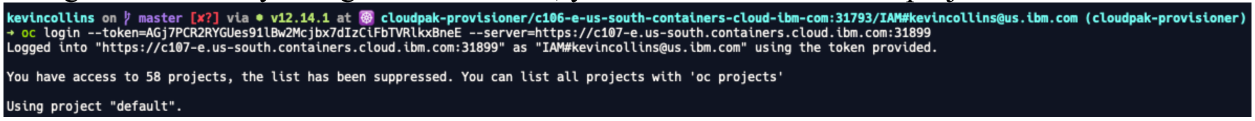
```
oc login --token=AGj7PCR2RYGUes91lBw2Mcjbx7dIzCiFbTVRlkxBneE --server=https://c107-e.us-south.containers.cloud.ibm.com:31899
```

Use this token directly against the API

```
curl -H "Authorization: Bearer AGj7PCR2RYGUes91lBw2Mcjbx7dIzCiFbTVRlkxBneE"
    "https://c107-e.us-south.containers.cloud.ibm.com:31899/apis/user.openshift.io/v1/users/~"
```

Copy the Log in with this token field.

Next, open up your iTerm2 terminal and paste the login command you just copied.
Being the first time you log into the cluster, you will be on the default project.



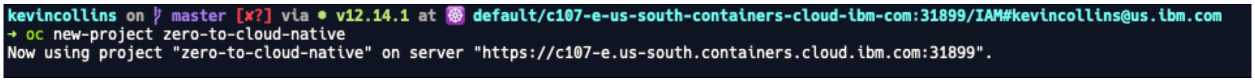
```
kevincollins on ᚠ master [x?] via ● v12.14.1 at ⎈ cloudpak-provisioner/c106-e-us-south-containers-cloud-ibm-com:31793/IAM#kevincollins@us.ibm.com (cloudpak-provisioner)
→ oc login --token=AGj7PCR2RYGUes91lBw2Mcjbx7dIzCiFbTVRlkxBneE --server=https://c107-e.us-south.containers.cloud.ibm.com:31899
Logged into "https://c107-e.us-south.containers.cloud.ibm.com:31899" as "IAM#kevincollins@us.ibm.com" using the token provided.

You have access to 58 projects, the list has been suppressed. You can list all projects with 'oc projects'

Using project "default".
```

Next, we need to create our zero-to-cloud-native project. In the terminal type:

```
oc new-project zero-to-cloud-native
```

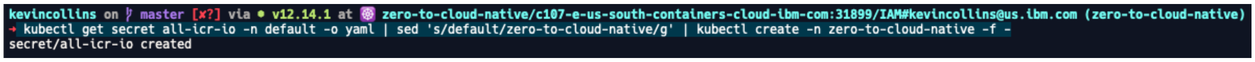


```
kevincollins on ᚠ master [x?] via ● v12.14.1 at ⎈ default/c107-e-us-south-containers-cloud-ibm-com:31899/IAM#kevincollins@us.ibm.com
→ oc new-project zero-to-cloud-native
Now using project "zero-to-cloud-native" on server "https://c107-e.us-south.containers.cloud.ibm.com:31899".
```

4 – 2 Create Image Pull Secrets

While we are in the terminal, we also need to copy the image pull secret from the default namespace to the zero-to-cloud-native namespace we just created. The image pull secret allows our cluster to pull images from the IBM Cloud Container Registry. When you create an Managed OpenShift Cluster an image pull secret named all-icr-io is created in the default namespace automatically. In order for to us the IBM Cloud Container Registry, we need to copy this secret to the new namespace we created. Enter the command below in your terminal to copy the image pull secret from the default namespace to the namespace we just created.

```
kubectl get secret all-icr-io -n default -o yaml | sed 's/default/zero-to-
cloud-native/g' | kubectl create -n zero-to-cloud-native -f -
```



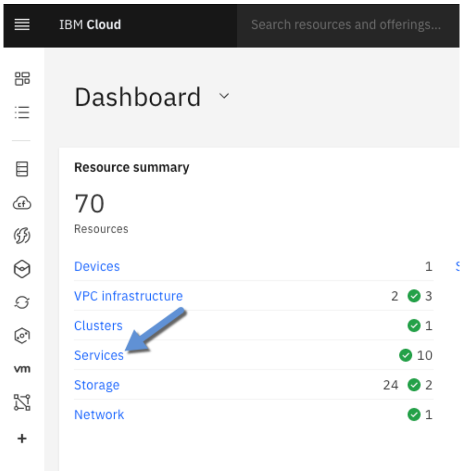
```
kevincollins on ᚠ master [x?] via ● v12.14.1 at ⎈ zero-to-cloud-native/c107-e-us-south-containers-cloud-ibm-com:31899/IAM#kevincollins@us.ibm.com (zero-to-cloud-native)
→ kubectl get secret all-icr-io -n default -o yaml | sed 's/default/zero-to-cloud-native/g' | kubectl create -n zero-to-cloud-native -f -
secret/all-icr-io created
```

This will create an image pull secret in our newly created namespace so that we can pull images from IBM Cloud Container Registry.

5 - Configure IBM Cloud Messages with RabbitMQ

Our zero-to-cloud-native application will use message queues to communicate between the frontend and backend microservices. We will need to create message queue that our microservices will use before we deploy our code.

From the dashboard view on IBM Cloud, click on Services.



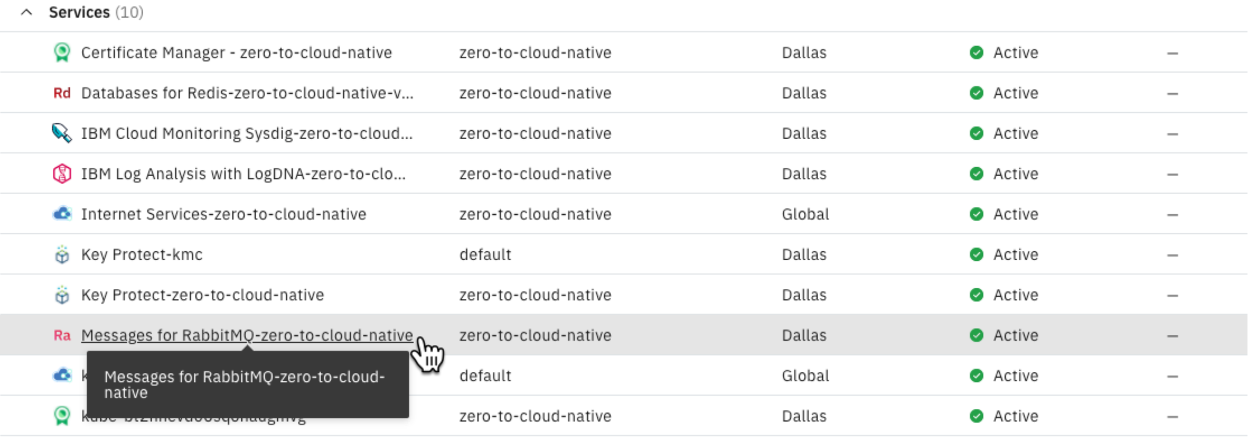
Dashboard

Resource summary

70
Resources

Devices	1	
VPC infrastructure	2	✓ 3
Clusters		✓ 1
Services		✓ 10
Storage	24	✓ 2
Network		✓ 1

Under services, find and click on your Messages for RabbitMQ instance.



Services (10)

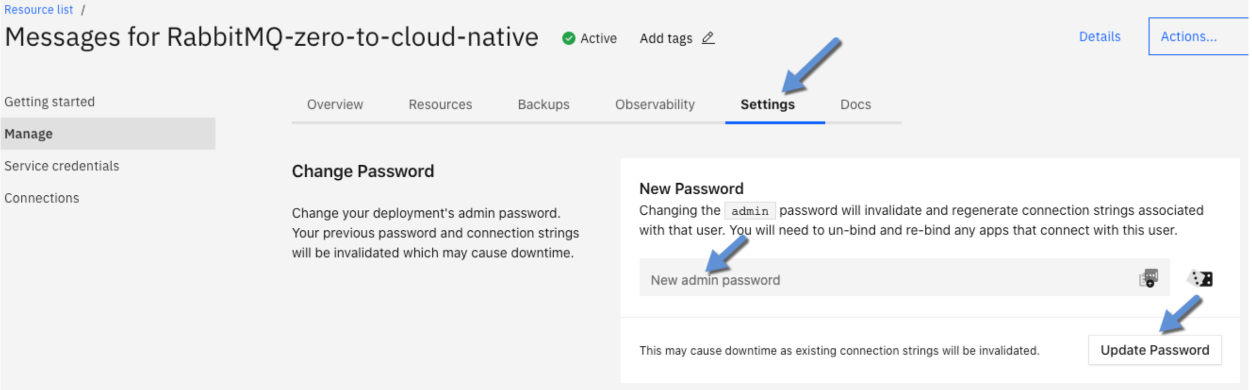
Certificate Manager - zero-to-cloud-native	zero-to-cloud-native	Dallas	✓ Active	–
Rd Databases for Redis-zero-to-cloud-native-v...	zero-to-cloud-native	Dallas	✓ Active	–
IBM Cloud Monitoring Sysdig-zero-to-cloud...	zero-to-cloud-native	Dallas	✓ Active	–
IBM Log Analysis with LogDNA-zero-to-clo...	zero-to-cloud-native	Dallas	✓ Active	–
Internet Services-zero-to-cloud-native	zero-to-cloud-native	Global	✓ Active	–
Key Protect-kmc	default	Dallas	✓ Active	–
Key Protect-zero-to-cloud-native	zero-to-cloud-native	Dallas	✓ Active	–
Ra Messages for RabbitMQ-zero-to-cloud-native	zero-to-cloud-native	Dallas	✓ Active	–
k	default	Global	✓ Active	–
kube-bt2nnevd0osqonaagnvg	zero-to-cloud-native	Dallas	✓ Active	–

Messages for RabbitMQ-zero-to-cloud-native

5 -1 Create an admin password

The first thing we need to do is generate an admin password for our Rabbit MQ Instance.

Click on settings, enter a good password, and click create Update Password.



Messages for RabbitMQ-zero-to-cloud-native

Change Password

Change your deployment's admin password. Your previous password and connection strings will be invalidated which may cause downtime.

New Password

Changing the `admin` password will invalidate and regenerate connection strings associated with that user. You will need to un-bind and re-bind any apps that connect with this user.

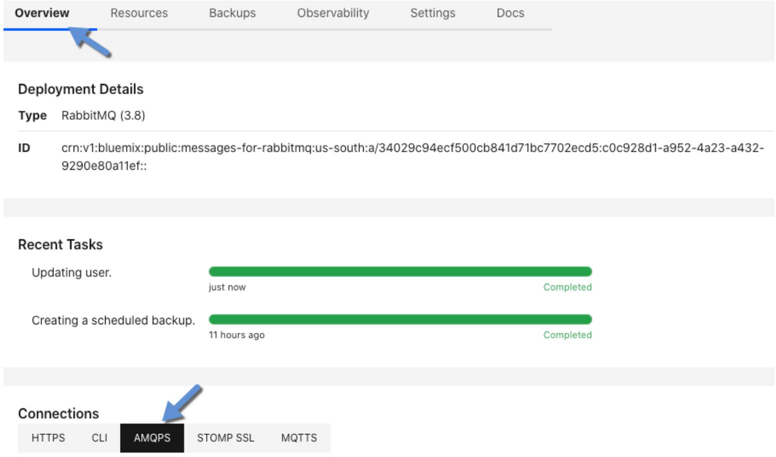
New admin password

This may cause downtime as existing connection strings will be invalidated.

IMPORTANT: Throughout this tutorial, there are several settings that you will need to remember for access later on. Create a temporary text file to store these settings. The first that you will need to remember is the **admin password for RabbitMQ**.

5 – 2 Rabbit MQ Certificate, Hostname and Port

Next, we will be storing the TLS certificate that we created for RabbitMQ in our instance of certificate manager that we are using to manage all of our certificates. To do so we will need to download the certificate. To do so, click back to the Overview view and click on the AMQPS Connection



Deployment Details

Type RabbitMQ (3.8)

ID crn:v1:bluemix:public:messages-for-rabbitmq:us-south:a/34029c94ecf500cb841d71bc7702ecd5:c0c928d1-a952-4a23-a432-9290e80a11ef::

Recent Tasks

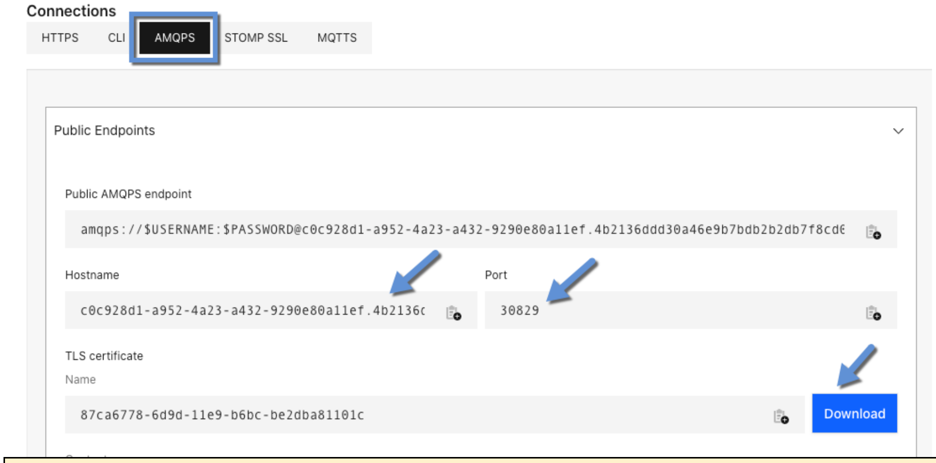
Updating user. just now Completed

Creating a scheduled backup. 11 hours ago Completed

Connections

.

Scroll down to TLS certificate section and click Download. While you on this screen, also note the Hostname and Port number as we will need this in the next section. Make sure you are on the AMQPS tab. Our application will use the Python Pika library which will use the AMQPS connection.



Public AMQPS endpoint

```
amqps://$USERNAME:$PASSWORD@c0c928d1-a952-4a23-a432-9290e80a11ef.4b2136ddd30a46e9b7bdb2b2db7f8cd6
```

Hostname: `c0c928d1-a952-4a23-a432-9290e80a11ef.4b2136c`

Port: `30829`

TLS certificate

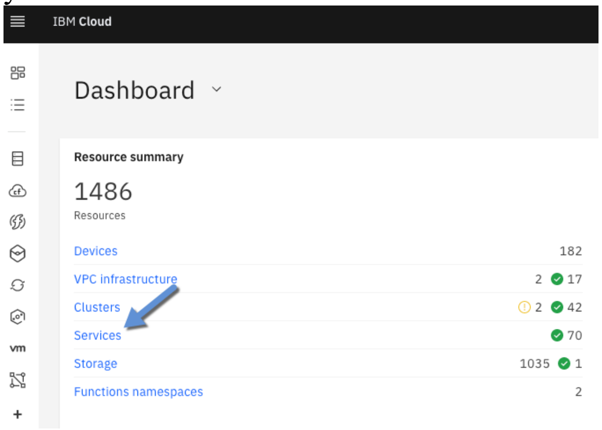
Name: `87ca6778-6d9d-11e9-b6bc-be2dba81101c`

IMPORTANT: add the RabbitMQ **Hostname** and **port** to your temporary file, you will need these values later.

Also note where you saved the certificate. Navigate to the directory where you saved the certificate file and rename the file so it ends with .pem.

5 – 3 Add RabbitMQ Certificate to Certificate Manager

Next, we will add the Messages for RabbitMQ certificate to Certificate Manager. Navigate to your IBM Cloud Dashboard and click on Services.

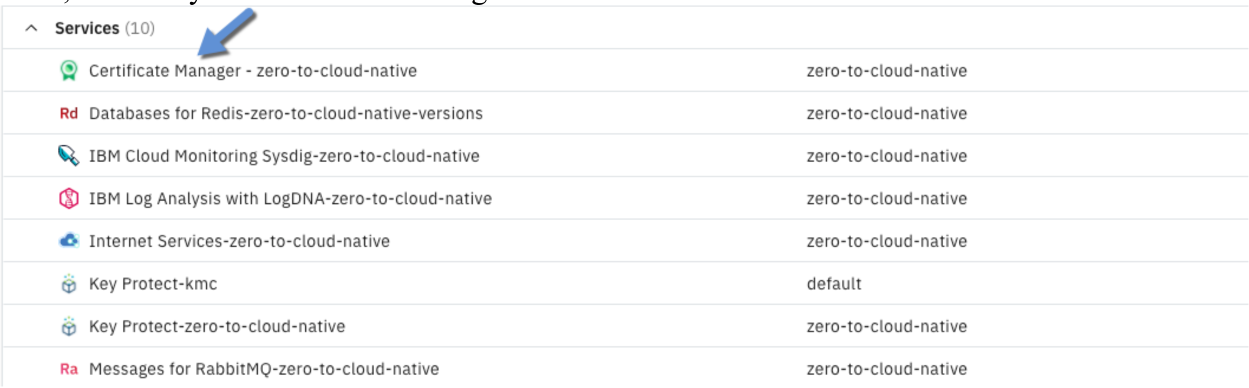


Resource summary

1486
Resources

Devices	182
VPC infrastructure	2 ✓ 17
Clusters	ⓘ 2 ✓ 42
Services	✓ 70
Storage	1035 ✓ 1
Functions namespaces	2

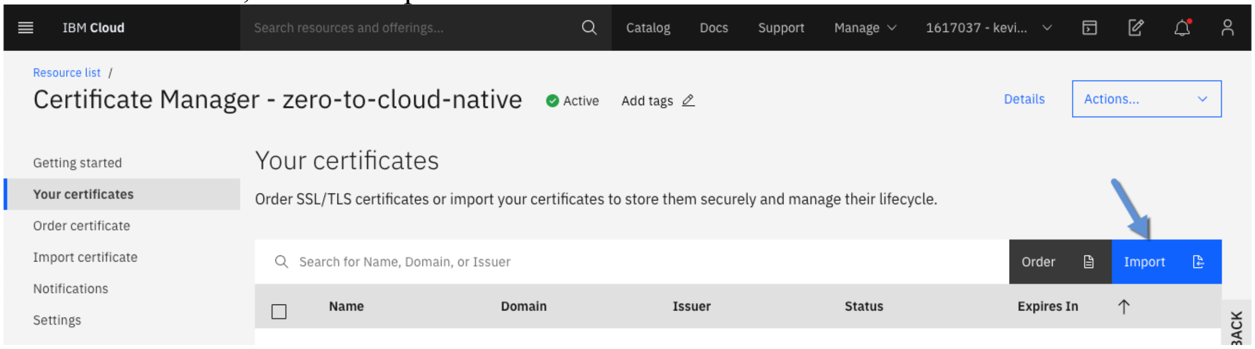
Next, click on your Certificate Manager instance.



Services (10)

Certificate Manager - zero-to-cloud-native	zero-to-cloud-native
Rd Databases for Redis-zero-to-cloud-native-versions	zero-to-cloud-native
IBM Cloud Monitoring Sysdig-zero-to-cloud-native	zero-to-cloud-native
IBM Log Analysis with LogDNA-zero-to-cloud-native	zero-to-cloud-native
Internet Services-zero-to-cloud-native	zero-to-cloud-native
Key Protect-kmc	default
Key Protect-zero-to-cloud-native	zero-to-cloud-native
Ra Messages for RabbitMQ-zero-to-cloud-native	zero-to-cloud-native

On the next screen, click on Import.



Certificate Manager - zero-to-cloud-native

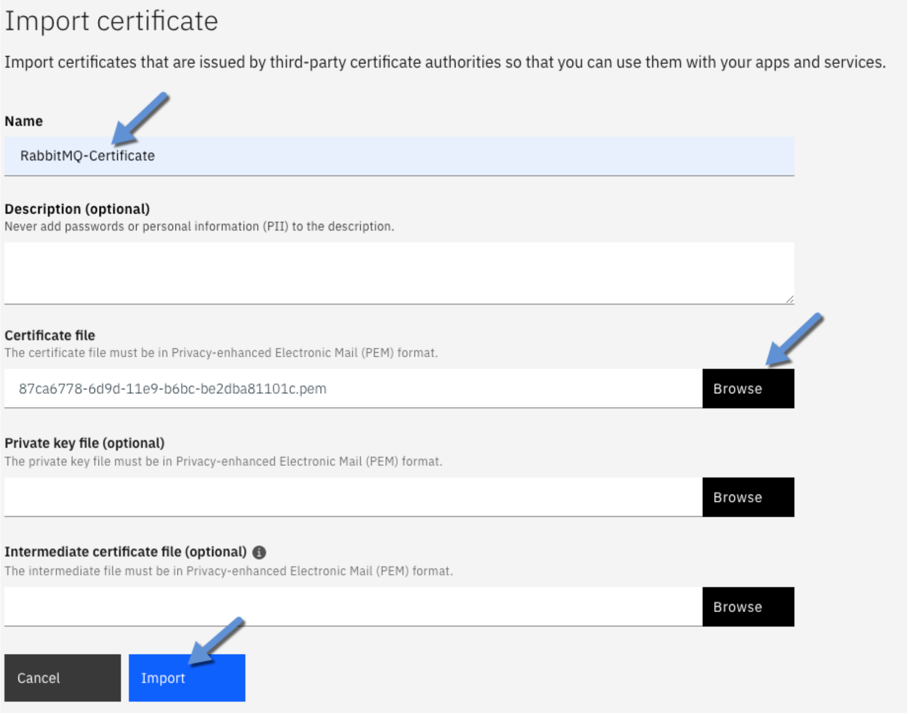
Your certificates

Order SSL/TLS certificates or import your certificates to store them securely and manage their lifecycle.

□	Name	Domain	Issuer	Status	Expires In ↑

Enter a name that so that you can find the certificate for RabbitMQ later. I will name the certificate RabbitMQ-Certificate.

Next, click browse to find the certificate file you downloaded and renamed in the previous step and click Import.



Import certificate

Import certificates that are issued by third-party certificate authorities so that you can use them with your apps and services.

Name
RabbitMQ-Certificate

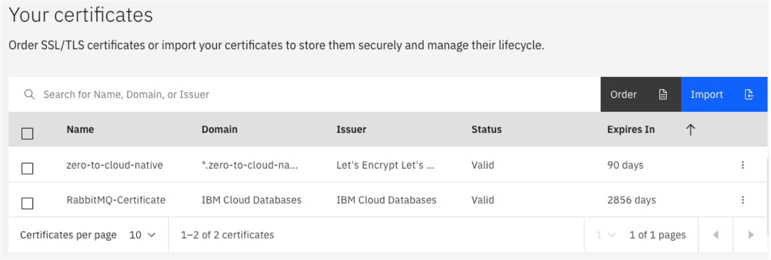
Description (optional)
Never add passwords or personal information (PII) to the description.

Certificate file
The certificate file must be in Privacy-enhanced Electronic Mail (PEM) format.
87ca6778-6d9d-11e9-b6bc-be2dba81101c.pem

Private key file (optional)
The private key file must be in Privacy-enhanced Electronic Mail (PEM) format.

Intermediate certificate file (optional)
The intermediate file must be in Privacy-enhanced Electronic Mail (PEM) format.

On the next page, you should see two certificates. One for the domain we created in Part 3, and the newly created RabbitMQ Certificate.

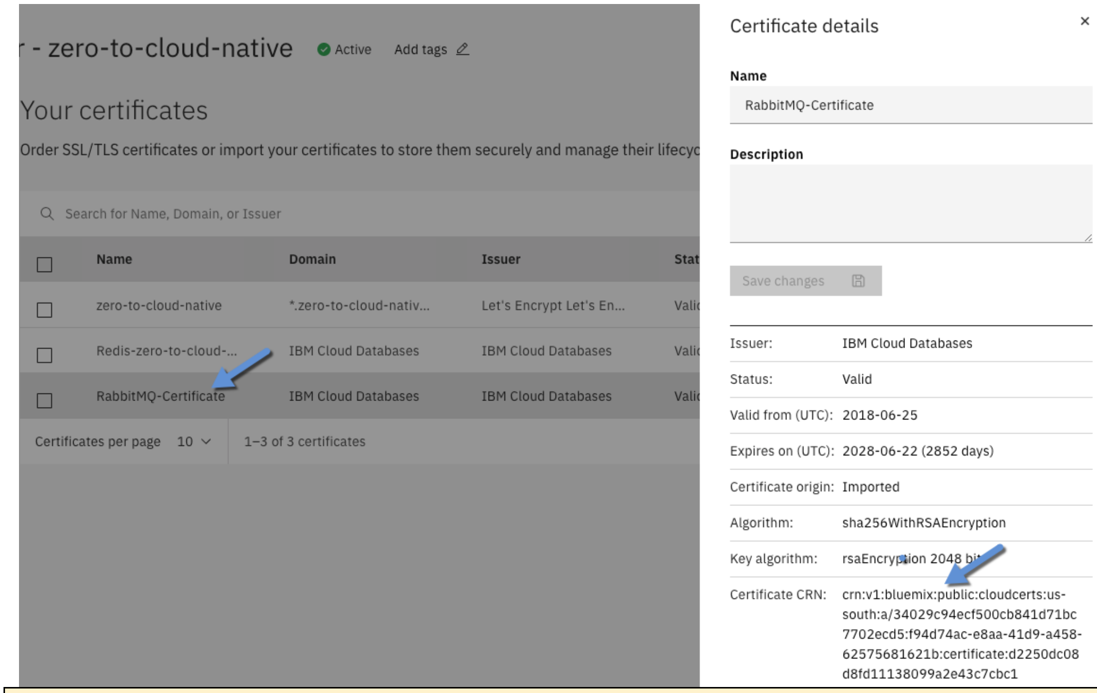


Your certificates

Order SSL/TLS certificates or import your certificates to store them securely and manage their lifecycle.

□	Name	Domain	Issuer	Status	Expires In ↑
□	zero-to-cloud-native	*.zero-to-cloud-na...	Let's Encrypt Let's ...	Valid	90 days
□	RabbitMQ-Certificate	IBM Cloud Databases	IBM Cloud Databases	Valid	2856 days

To access the contents of the certificate later on from Certificate Manager, we will need to use the Certificate ID or CRN. Click on the name you gave for the RabbitMQ Certificate. This will bring up a second window that will show the Certificate CRN.


r - zero-to-cloud-native
Active
Add tags
Your certificates
Order SSL/TLS certificates or import your certificates to store them securely and manage their lifecyc
Search for Name, Domain, or Issuer

	Name	Domain	Issuer	Stat
□	zero-to-cloud-native	*.zero-to-cloud-nativ...	Let's Encrypt Let's En...	Vali
□	Redis-zero-to-cloud-...	IBM Cloud Databases	IBM Cloud Databases	Vali
□	RabbitMQ-Certificate	IBM Cloud Databases	IBM Cloud Databases	Vali

Certificates per page 10
1–3 of 3 certificates
Certificate details
×

Name: RabbitMQ-Certificate

Description:

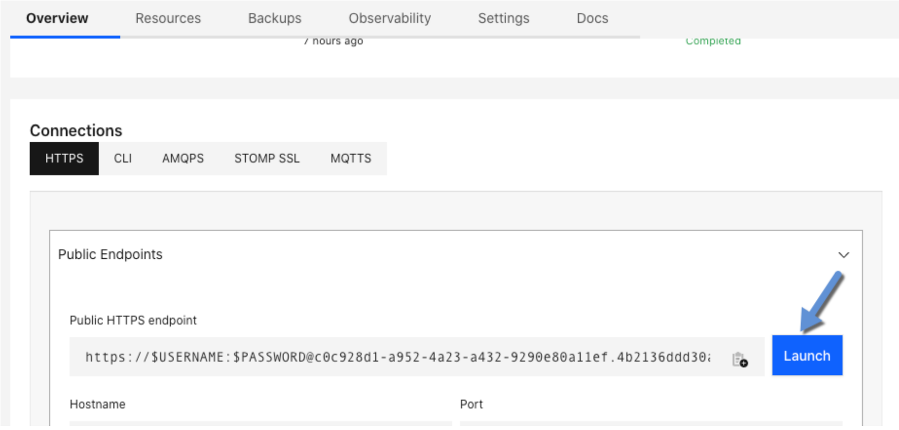
Issuer:	IBM Cloud Databases
Status:	Valid
Valid from (UTC):	2018-06-25
Expires on (UTC):	2028-06-22 (2852 days)
Certificate origin:	Imported
Algorithm:	sha256WithRSAEncryption
Key algorithm:	rsaEncryption 2048 bit
Certificate CRN:	crn:v1:bluemix:public:cloudcerts:us-south:a/34029c94ecf500cb841d71bc7702ecd5:f94d74ac-e8aa-41d9-a458-62575681621b:certificate:d2250dc08d8fd11138099a2e43c7cbc1

IMPORTANT: add the **RabbitMQ Certificate CRN** to your temporary file, you will need this value later.

5 – 4 – Create RabbitMQ Queue

Next, we need to create our RabbitMQ Queue. The microservices that we will be using a queue are '`get OpenShift tokens`' and '`get OpenShift versions`'. Using the domain driven model, we can view both of these microservices as an 'OpenShift' domain. As these are both very quick realtime APIs, we can combine them together in a single microservice sharing a single queue.

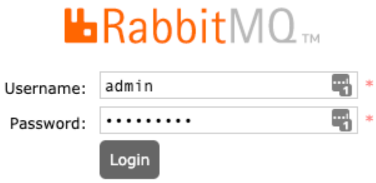
To create a queue, navigate back to your Messages by RabbitMQ service. Scrolling down to the Connections section, click on Launch next to the Public HTTPS endpoint.



Public HTTPS endpoint

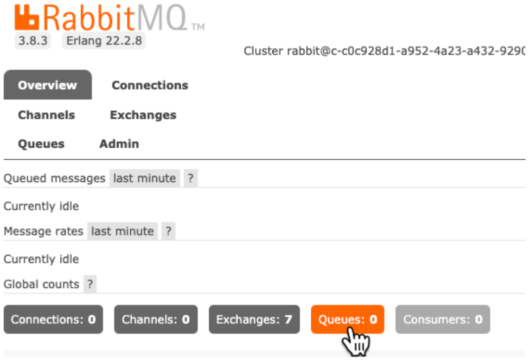
```
https://$USERNAME:$PASSWORD@c0c928d1-a952-4a23-a432-9290e80a11ef.4b2136ddd30a
```

On the next login screen, you will get a warning that the site is not secure. Just ignore that and continue to the site. Enter the admin username and password you saved from the earlier step.

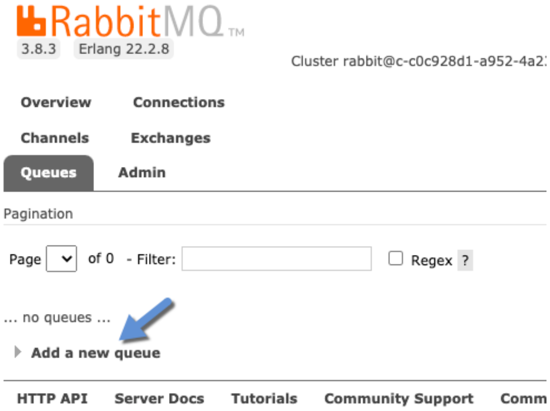

Username: admin

Password: •••••••••

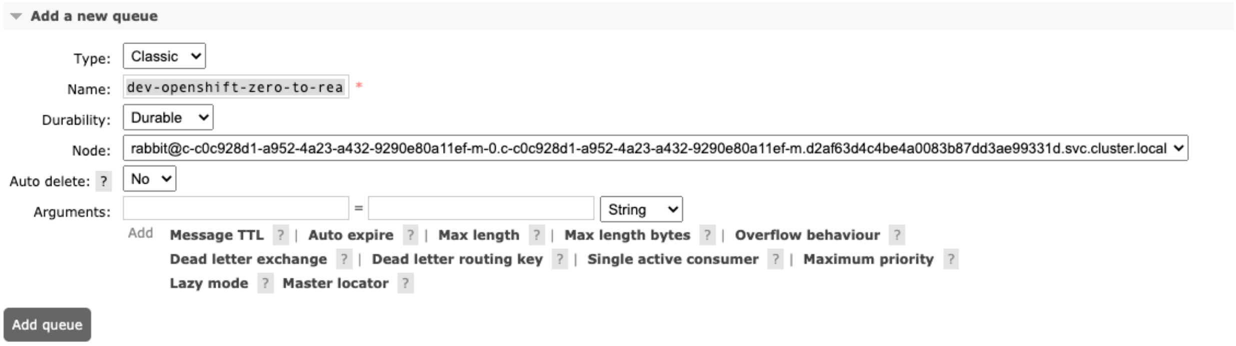
On the next page, click on Queues.



Next, click on Add a new queue.



On the next screen, you need to give your queue a name. I'm going to treat this environment as a dev environment so taking into consideration which microservice this queue will service, I am going to name the queue `dev-ocp-realtime-02cn`



Add a new queue

Type: Classic

Name: dev-openshift-zero-to-rea *

Durability: Durable

Node: rabbit@c-c0c928d1-a952-4a23-a432-9290e80a11ef-m-0.c-c0c928d1-a952-4a23-a432-9290e80a11ef-m.d2af63d4c4be4a0083b87dd3ae99331d.svc.cluster.local

Auto delete: ? No

Arguments: = String

Add Message TTL ? | Auto expire ? | Max length ? | Max length bytes ? | Overflow behaviour ?
Dead letter exchange ? | Dead letter routing key ? | Single active consumer ? | Maximum priority ?
Lazy mode ? Master locator ?
Add queue

After entering a name, click on Add Queue.

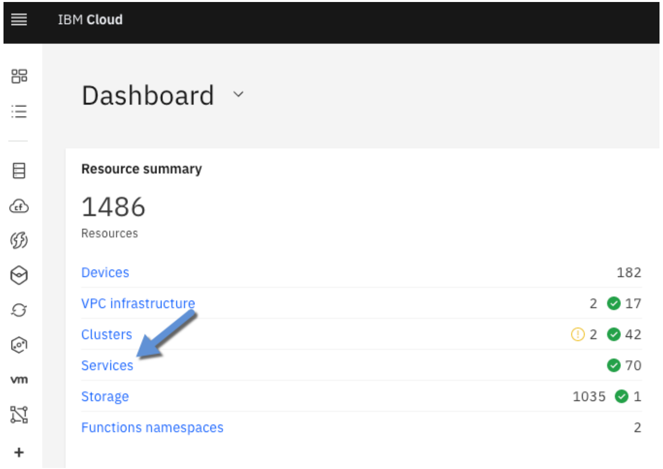
IMPORTANT: add the **RabbitMQ Queue** name to your temporary file, you will need this value later.

That finishes the setup and configuration of Messages by RabbitMQ! Overall, a very quick and easy process for setting up RabbitMQ. The service not only provisioned an instance of Rabbit MQ, but is also encrypting all the data on queues with an encryption key we create in part 3 of the series, and will automatically we be backed up with encryption.

6 - Configure IBM Cloud Databases for Redis

Our cloud-native application architecture will use Redis as a caching database to store versions of OpenShift supported by IBM Cloud. To use Redis in our application, we will need to get the TLS certificate for our service and store it in our certificate manager instance.

From the IBM Cloud dashboard, click on Services.

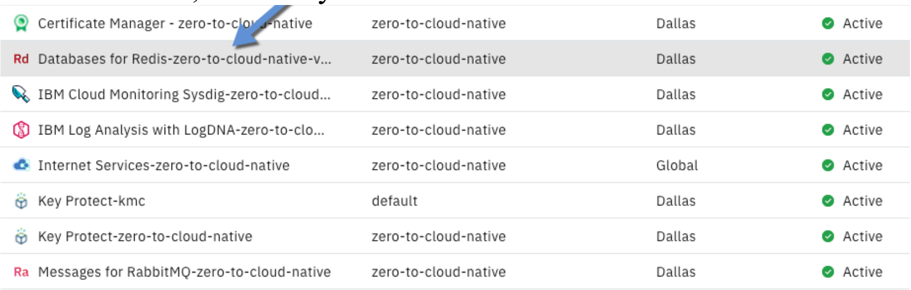


Resource summary

1486
Resources

Devices	182
VPC infrastructure	2 ✓ 17
Clusters	ⓘ 2 ✓ 42
Services	✓ 70
Storage	1035 ✓ 1
Functions namespaces	2

Under services, click on your instance of Databases for Redis.

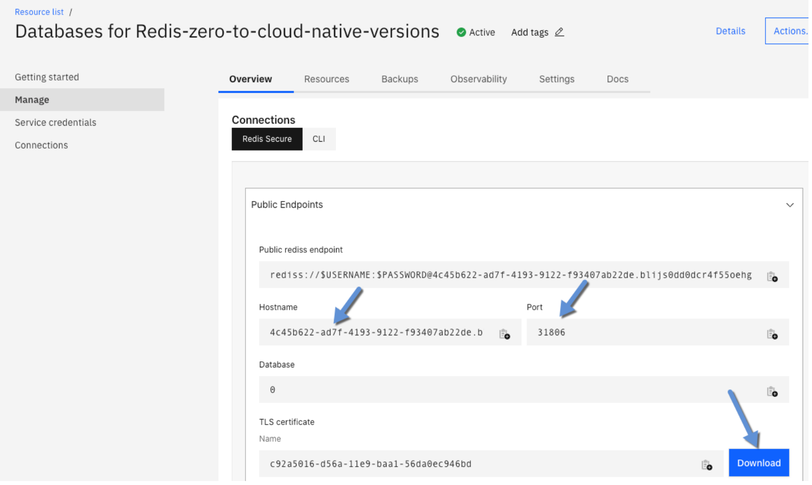


Certificate Manager - zero-to-cloud-native	zero-to-cloud-native	Dallas	✓ Active
Rd Databases for Redis-zero-to-cloud-native-v...	zero-to-cloud-native	Dallas	✓ Active
IBM Cloud Monitoring Sysdig-zero-to-cloud...	zero-to-cloud-native	Dallas	✓ Active
IBM Log Analysis with LogDNA-zero-to-clo...	zero-to-cloud-native	Dallas	✓ Active
Internet Services-zero-to-cloud-native	zero-to-cloud-native	Global	✓ Active
Key Protect-kmc	default	Dallas	✓ Active
Key Protect-zero-to-cloud-native	zero-to-cloud-native	Dallas	✓ Active
Ra Messages for RabbitMQ-zero-to-cloud-native	zero-to-cloud-native	Dallas	✓ Active

6-1 Obtain Redis Certificate, Hostname and Port

After clicking on the service, you will see an Overview screen for the service. As we did with the RabbitMQ, scroll down to the TLS certificate and click Download.

IMPORTANT: add the Redis **Hostname** and **Port number** to your temporary file, you will need these values later.



Databases for Redis-zero-to-cloud-native-versions

Connections

Public Endpoints

Public rediss endpoint

```
rediss://$USERNAME:$PASSWORD@4c45b622-ad7f-4193-9122-f93407ab22de.blijs0dd0dcr4f55oehg
```

Hostname: `4c45b622-ad7f-4193-9122-f93407ab22de.b`

Port: `31806`

Database: `0`

TLS certificate

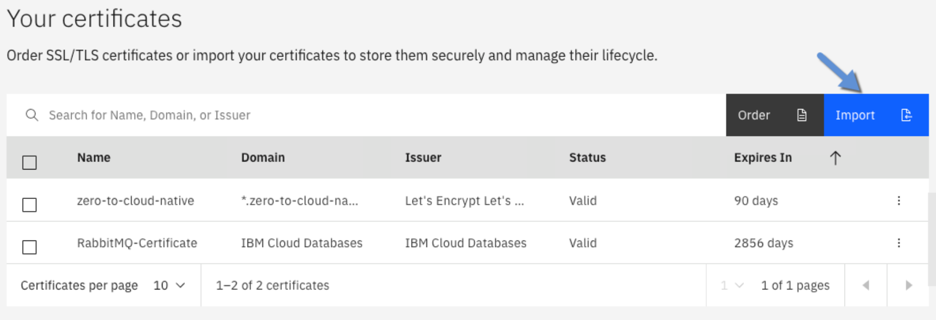
Name: `c92a5016-d56a-11e9-baa1-56da0ec946bd`

Navigate to where you saved the certificate, and like we did with RabbitMQ, rename the certificate so is ends with .pem.

6 – 2 Add Redis Certificate to Certificate Manager

Now that we have the certificate, we need to store the certificate in our certificate manage instance.

Going back to certificate manager, click on Import.



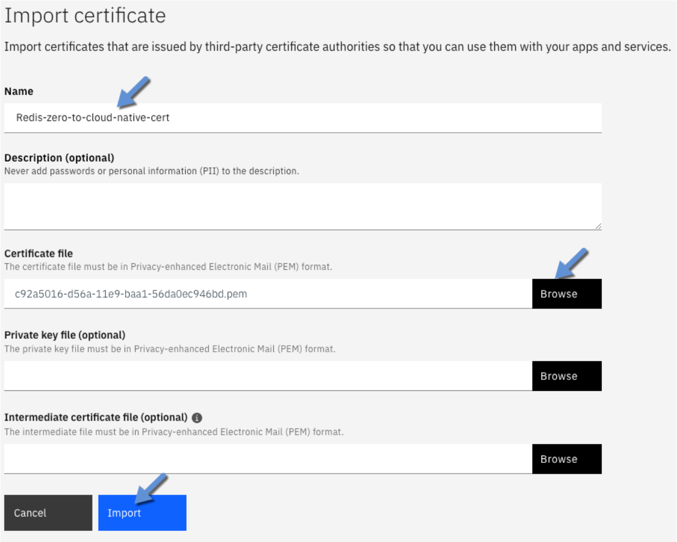
Your certificates

Order SSL/TLS certificates or import your certificates to store them securely and manage their lifecycle.

☐	Name	Domain	Issuer	Status	Expires In ↑
☐	zero-to-cloud-native	*.zero-to-cloud-na...	Let's Encrypt Let's ...	Valid	90 days
☐	RabbitMQ-Certificate	IBM Cloud Databases	IBM Cloud Databases	Valid	2856 days

Certificates per page 10 ∨ 1–2 of 2 certificates 1 ∨ 1 of 1 pages

Give your certificate a meaningful name. I will be using `Redis-zero-to-cloud-native-cert`. Select the certificate file you downloaded and renamed and click Import.



Import certificate

Import certificates that are issued by third-party certificate authorities so that you can use them with your apps and services.

Name

Redis-zero-to-cloud-native-cert

Description (optional)

Never add passwords or personal information (PII) to the description.

Certificate file

The certificate file must be in Privacy-enhanced Electronic Mail (PEM) format.

c92a5016-d56a-11e9-baa1-56da0ec946bd.pem

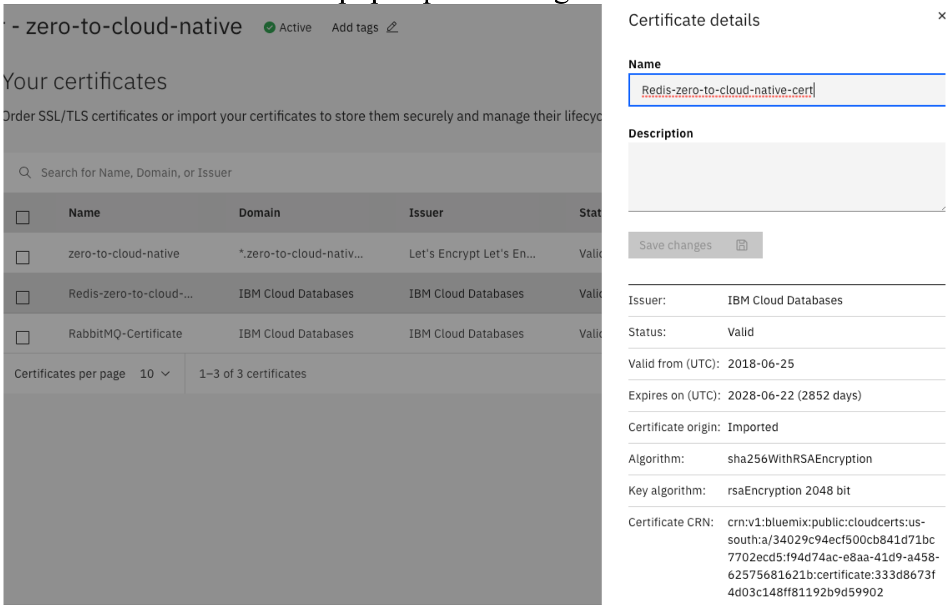
Private key file (optional)

The private key file must be in Privacy-enhanced Electronic Mail (PEM) format.

Intermediate certificate file (optional)

The intermediate file must be in Privacy-enhanced Electronic Mail (PEM) format.

As we did with the Rabbit MQ certificate, we need to copy the certificate CRN for our Redis certificate. Click on the Redis certificate row in certificate manager and then copy the certificate CRN value the screen the pops up on the right-hand side.



- zero-to-cloud-native

Your certificates

Order SSL/TLS certificates or import your certificates to store them securely and manage their lifecyc

Search for Name, Domain, or Issuer

□	Name	Domain	Issuer	Stat
□	zero-to-cloud-native	*.zero-to-cloud-nativ...	Let's Encrypt Let's En...	Vali
□	Redis-zero-to-cloud-...	IBM Cloud Databases	IBM Cloud Databases	Vali
□	RabbitMQ-Certificate	IBM Cloud Databases	IBM Cloud Databases	Vali

Certificates per page 10 ⌄ 1–3 of 3 certificates

Certificate details

Name

Redis-zero-to-cloud-native-cert

Description

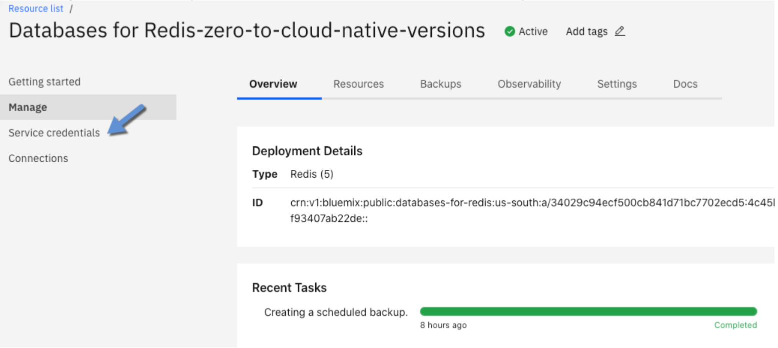
Issuer:	IBM Cloud Databases
Status:	Valid
Valid from (UTC):	2018-06-25
Expires on (UTC):	2028-06-22 (2852 days)
Certificate origin:	Imported
Algorithm:	sha256WithRSAEncryption
Key algorithm:	rsaEncryption 2048 bit
Certificate CRN:	crn:v1:bluemix:public:cloudcerts:us-south:a/34029c94ecf500cb841d71bc7702ecd5:f94d74ac-e8aa-41d9-a458-62575681621b:certificate:333d8673f4d03c148ff81192b9d59902

IMPORTANT: add the **Redis Certificate** CRN to your temporary file, you will need this value later.

6 – 3 Create Redis Service Credentials

We also need to create service credentials for our Redis Instance.

Going back to your Redis instance from your resource list, click on Service Credentials.



Resource list /

Databases for Redis-zero-to-cloud-native-versions ● Active Add tags

Deployment Details

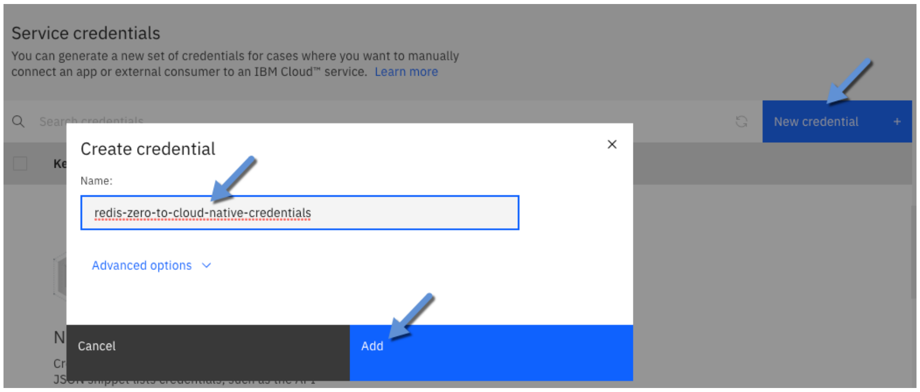
Type Redis (5)

ID crn:v1:bluemix:public:databases-for-redis:us-south:a/34029c94ecf500cb841d71bc7702ecd5:4c45l
f93407ab22de::

Recent Tasks

Creating a scheduled backup. 8 hours ago Completed

On the next screen, click New Credential, give your service credentials a good name, and click Add.



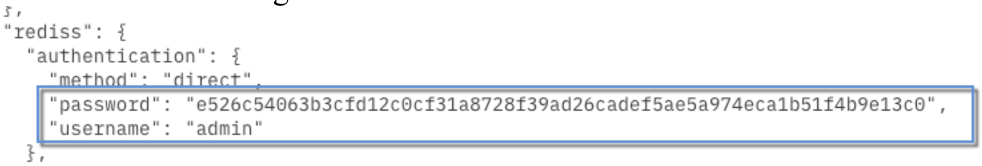
Service credentials

You can generate a new set of credentials for cases where you want to manually connect an app or external consumer to an IBM Cloud™ service. Learn more

Create credential

Name: redis-zero-to-cloud-native-credentials

Expand the service credentials and scroll down to the authentication section and note the username and password. Make sure to store this user id and password in a place you will remember as we will be using it in the next section.



```
},
"rediss": {
  "authentication": {
    "method": "direct",
    "password": "e526c54063b3cfd12c0cf31a8728f39ad26cadef5ae5a974eca1b51f4b9e13c0",
    "username": "admin"
  },
```

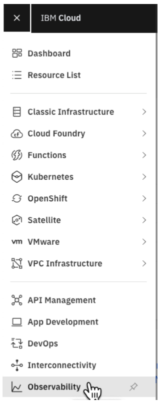
IMPORTANT: add the Redis **username** (should be admin), and **password** to your temporary file, you will need these values later.

And that is it, Redis is ready to be used.

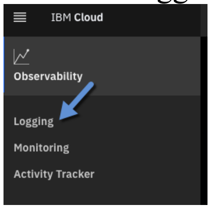
7 - Configure App to Log with LogDNA

There will be a future part of this series that will go through a deep dive in using LogDNA. However, in able for our application to send meaningful messages to LogDNA, there are a couple of things we need to do.

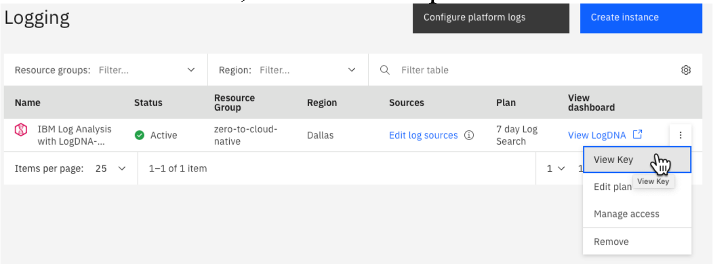
First, we need to retrieve our LogDNA API key. From the IBM Cloud hamburger menu, select Observability.



Click on Logging.



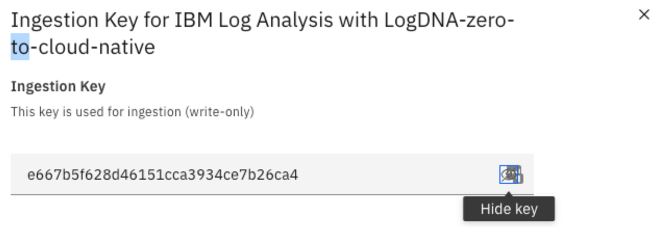
On the next screen, click on the options icon and select view Key.



Name	Status	Resource Group	Region	Sources	Plan	View dashboard
IBM Log Analysis with LogDNA-...	Active	zero-to-cloud-native	Dallas	Edit log sources	7 day Log Search	View LogDNA

Click on show key and copy the key to a temporary file that we will use later.

IMPORTANT: add the LogDNA Key to your temporary file, you will need these values later.



Ingestion Key for IBM Log Analysis with LogDNA-zero-to-cloud-native

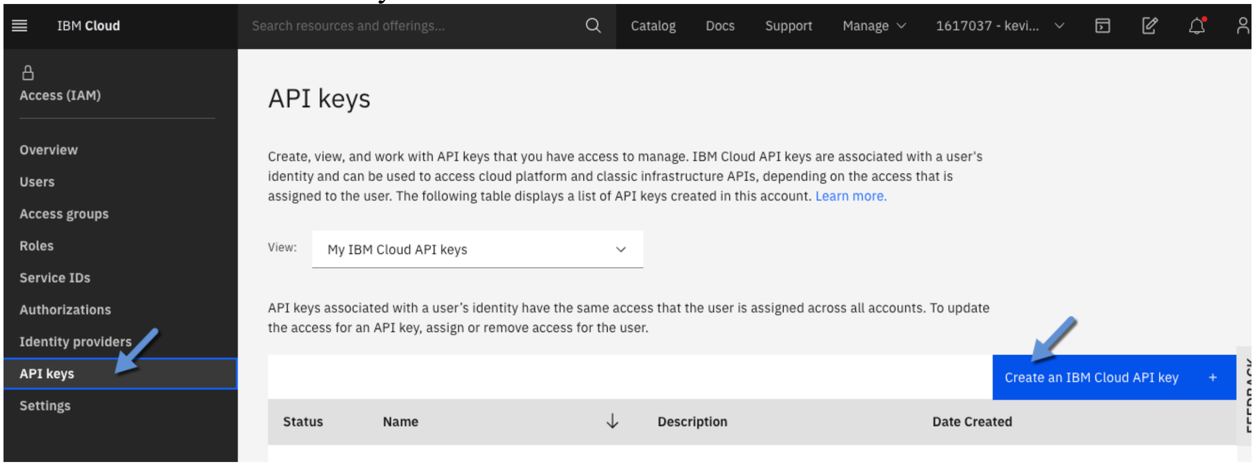
Ingestion Key

This key is used for ingestion (write-only)

e667b5f628d46151cca3934ce7b26ca4

8 - Create an IBM Cloud API Key

If you don't already have an API Key, you will need to create one. From the IBM Cloud Console, select Manage and then Access (IAM). On the next screen, click API keys and then Create an IBM Cloud API key.



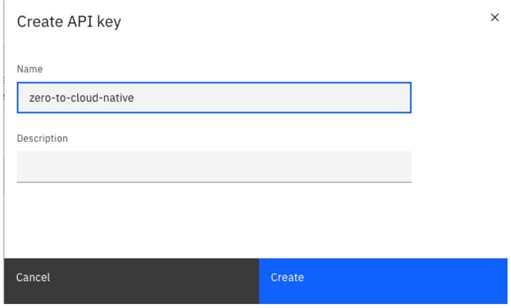
API keys

Create, view, and work with API keys that you have access to manage. IBM Cloud API keys are associated with a user's identity and can be used to access cloud platform and classic infrastructure APIs, depending on the access that is assigned to the user. The following table displays a list of API keys created in this account. Learn more.

View: My IBM Cloud API keys

API keys associated with a user's identity have the same access that the user is assigned across all accounts. To update the access for an API key, assign or remove access for the user.

On the next screen, give your API Key a name and click create copy and paste it in a file that you won't lose. We will be using this API key throughout the tutorial so make sure to store it somewhere so that you won't lose it.   Do not share this API key with anyone as the API key will allow whoever has it do pretty much anything to your IBM Cloud account.



Create API key

Name

zero-to-cloud-native

Description

IMPORTANT: add the API Key you just created to your temporary file, you will need this value later.

9 - Container Registry

Before we can deploy our images into a container registry, we will need to create a namespace in IBM Cloud to store our images. To create a container registry namespace, start iTerm2 and login into IBM Cloud with the following command.

```
ic login --apikey=<apikey you copied above>
```

Next, we need to log into the container registry.

```
ic cr login
```

Now, we can create a new container registry namespace.

```
ic cr namespace-add zero-to-cloud-native
```

Note that you cannot have the same container registry namespace as someone else. You will need to create a unique name, suggestion is to use zero-to-cloud-native + your initials

IMPORTANT: note the namespace you just create in your temporary file, you will need this value later.

That completes the setup and configuration of all the services our application will use, next up we will go through the code base and then deploy the application.