Integrate Slack and PagerDuty

Objective

Bluemix DevOps services include toolchains working together to support your dev elopment and deployment tasks. The integrations across a toolchain help ease the workflows between the tools so they work in synergy. This lab demonstrates how to set-up and use some of the Bluemix toolchains with a variety of industry tools such as GitHub, Slack and PagerDuty to make your move to DevOps easier. **Note:** Toolchains are currently available in the US South region only and the instructions in this lab are written for the US South region.

This lab uses the simple toolchain for the Orders API microservice.

Teaming

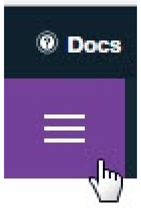
Software development is a team activity. The lab scenario also shows how Bluemix Continuous Delivery tool integrations can be used to alerts teams when activities occur (such as builds or deployments) as well as when critical incidents happen (such as a build failing or an application outage).

- Slack is configured to alert the team when activities occur
- PagerDuty is configured to alert the team about critical incidents

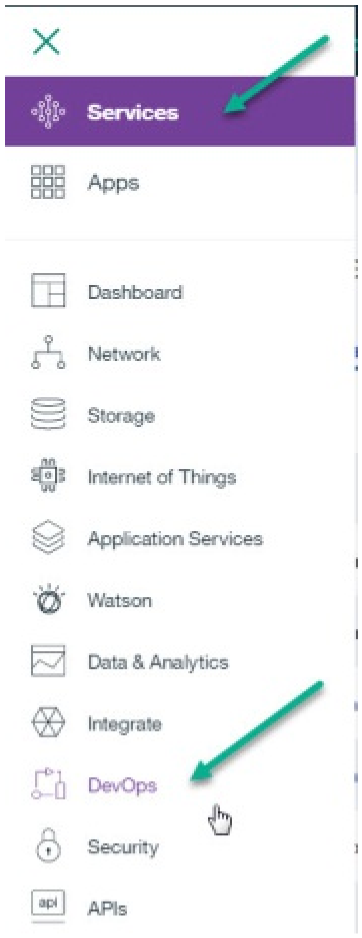
Tasks

Task 1: Integrate Slack into Toolchain

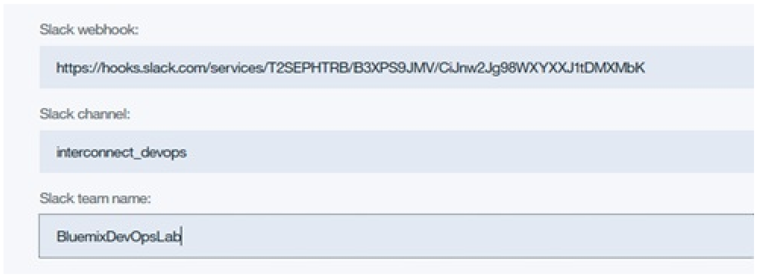
The visual display of the Delivery Pipeline is great if you want to watch it. But if you don't want to watch or if you did not even start the build, how do you get notified of events that you are interested in? One way is Slack. Slack provides real-time messaging for team communications. You can integrate Slack with your Bluemix DevOps Services project so that notifications about build results from your Delivery Pipeline are posted on a Slack channel.

1. If you are not on the Toolchains page (if you don't see a button called *Create a Toolchain*), cick on the **Bluemix menu bar** in the upper left corner.
   

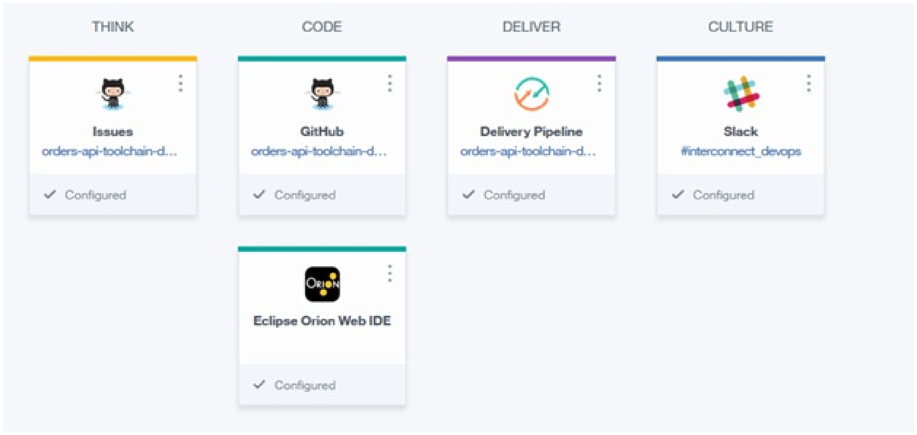
   and click on **Services** then **DevOps**



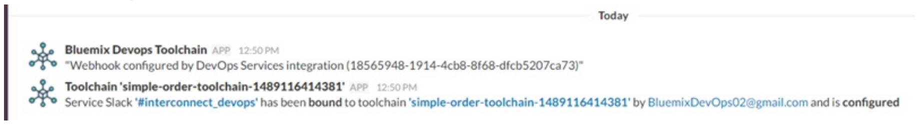
2. On the **simple-order-toolchain-*timestamp*** Toolchain page, click **Add a Tool**.
3. Select **Slack**
4. On the Slack Configuration page:
    - Enter the following all as one string as the Slack webhook. **https://hooks.slack.com/services/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**
    - Enter **#interconnect_devops** as the Slack channel.
    - Enter **BluemixDevOpsLab** as the Slack team name.


Slack webhook:
https://hooks.slack.com/services/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
Slack channel:
interconnect_devops
Slack team name:
BluemixDevOpsLab

5. Click **Create Integration**
6. The integration with Slack is added.



7. Switch to the browser tab that has Slack open. Slack notifies us that the integration was added.


Today

Bluemix Devops Toolchain APP 12:50 PM
"Webhook configured by DevOps Services integration (18565948-1914-4cb8-8f68-dfcb5207ca73)"

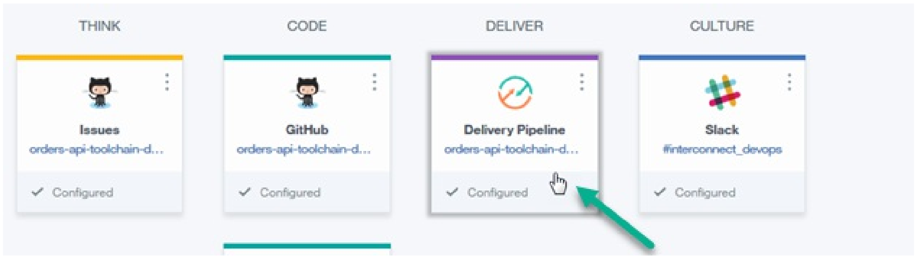
Toolchain 'simple-order-toolchain-1489116414381' APP 12:50 PM
Service Slack '#interconnect_devops' has been **bound** to toolchain 'simple-order-toolchain-1489116414381' by BluemixDevOps02@gmail.com and is **configured**

Task 2: Test Slack Integration

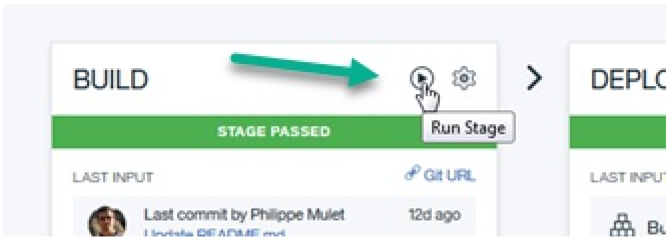
Looking at the *interconnect_devops* Slack channel, we see that other team members have been busy. Let's see what happens when we initiate a build.

1. Switch to the browser tab displaying the Bluemix console. It should be displaying the **simple-order-toolchain-*timestamp*** Toolchain page. If not, click on the Bluemix menu bar in the upper left corner, click on Services then DevOps. Click on Toolchains. Then click on **simple-order-toolchain-*timestamp***.
2. Click on the **Delivery Pipeline** tile for the orders-api-toolchain-demo

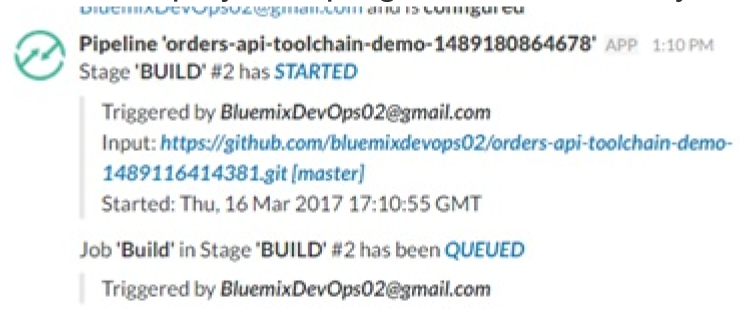
Delivery Pipeline.



3. Start an application build by clicking on the *BUILD* stage **Run Stage** icon.



4. Switching to the Slack browser tab, the Slack bluemixdevopslab channel displays the progress of this activity.



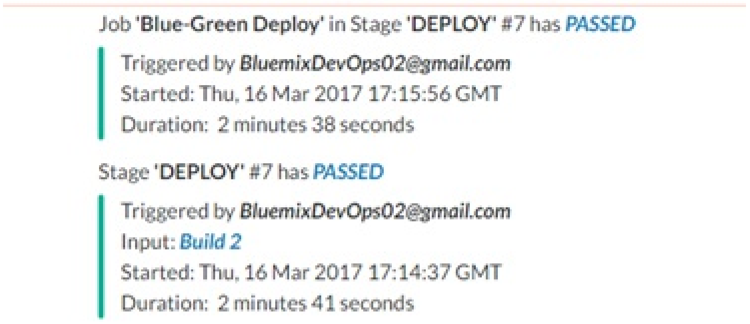
Pipeline 'orders-api-toolchain-demo-1489180864678' APP 1:10 PM
Stage **'BUILD'** #2 has *STARTED*

Triggered by *BluemixDevOps02@gmail.com*
Input: *https://github.com/bluemixdevops02/orders-api-toolchain-demo-1489116414381.git [master]*
Started: Thu, 16 Mar 2017 17:10:55 GMT

Job **'Build'** in Stage **'BUILD'** #2 has been *QUEUED*

Triggered by *BluemixDevOps02@gmail.com*

5. And the completion.



Job 'Blue-Green Deploy' in Stage 'DEPLOY' #7 has *PASSED*

Triggered by *BluemixDevOps02@gmail.com*
Started: Thu, 16 Mar 2017 17:15:56 GMT
Duration: 2 minutes 38 seconds

Stage 'DEPLOY' #7 has *PASSED*

Triggered by *BluemixDevOps02@gmail.com*
Input: *Build 2*
Started: Thu, 16 Mar 2017 17:14:37 GMT
Duration: 2 minutes 41 seconds

6. Through the Bluemix Continuous Delivery integration with Slack, teams get notified of events in the Delivery Pipeline.
7. Switch to the browser tab displaying the Bluemix console. Click the blue arrow in the upper left hand corner to return to the Toolchain.

Task 6: Integrate PagerDuty into Toolchain

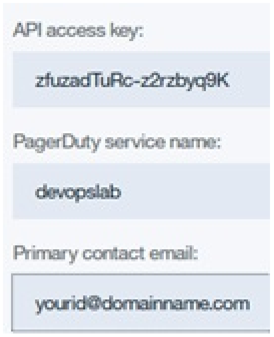
As a developer or operations person, you may want to be notified of the status of critical incidents. One way to be alerted is to integrate third-party tools such as PagerDuty. IBM Bluemix has made it easy to add PagerDuty integration into a Toolchain so the teams such as development or operations are notified of the status of deployments.

Note: The PagerDuty integration feature is not created by, affiliated with, or supported by PagerDuty, Inc.

1. On the **simple-order-toolchain-*timestamp*** Toolchain page, click **Add a Tool**.
2. Select **PagerDuty**
3. On the PagerDuty Configuration page:
    - Enter zfuzadTuRc-z2rzbyq9K as the API access key
    - Enter *devopslab* as the PagerDuty service name
    - Enter an email address you can access. Leave the phone number blank. PagerDuty has a place for you to enter your phone number if you want to receive text messages or

receive phone calls.

4. Click **Create Integration**



API access key:

zfuzadTuRc-z2rzbyq9K

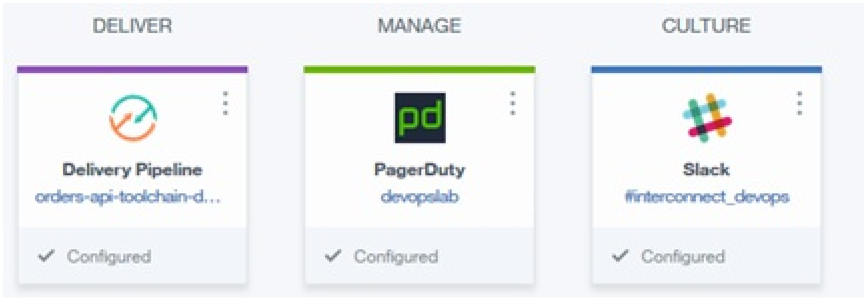
PagerDuty service name:

devopslab

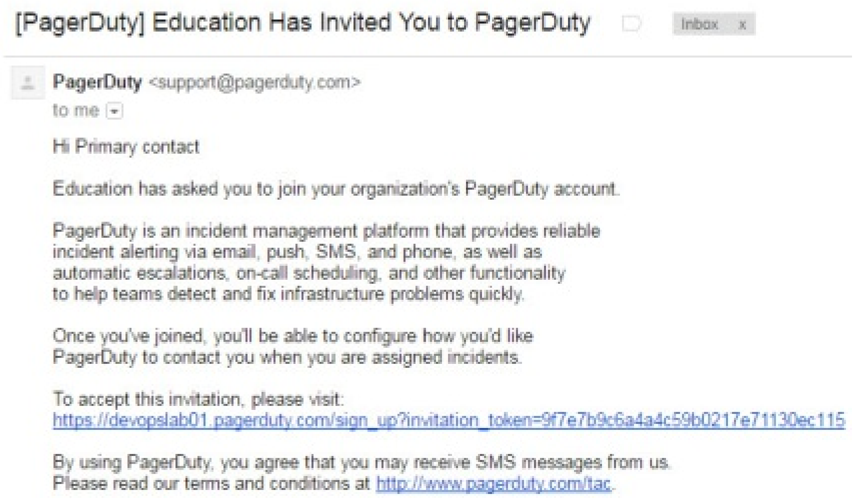
Primary contact email:

yourid@domainname.com

5. The integration is added.



6. If you want to see what PagerDuty sends, you need to access the eMail account you specified to accept the PagerDuty invitation.



[PagerDuty] Education Has Invited You to PagerDuty Inbox x

PagerDuty <support@pagerduty.com>
to me

Hi Primary contact

Education has asked you to join your organization's PagerDuty account.

PagerDuty is an incident management platform that provides reliable incident alerting via email, push, SMS, and phone, as well as automatic escalations, on-call scheduling, and other functionality to help teams detect and fix infrastructure problems quickly.

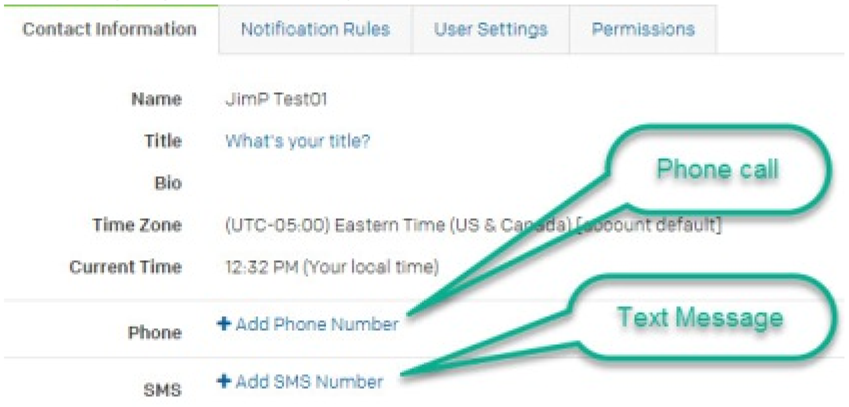
Once you've joined, you'll be able to configure how you'd like PagerDuty to contact you when you are assigned incidents.

To accept this invitation, please visit:
https://devopslab01.pagerduty.com/sign_up?invitation_token=9f7e7b9c6a4a4c59b0217e71130ec115

By using PagerDuty, you agree that you may receive SMS messages from us.
Please read our terms and conditions at http://www.pagerduty.com/tac.

7. If you click the link to accept the invitation, you will need to create a PagerDuty account associated with the eMail you specified. The lab has screenshots if you would prefer to not get the eMails. You can also

enter your phone information into your PagerDuty account. If you do this, you will get text messages and/or phone calls, perhaps a lot, depending on where you entered your phone number. Normal messaging rates apply.


Contact Information | Notification Rules | User Settings | Permissions

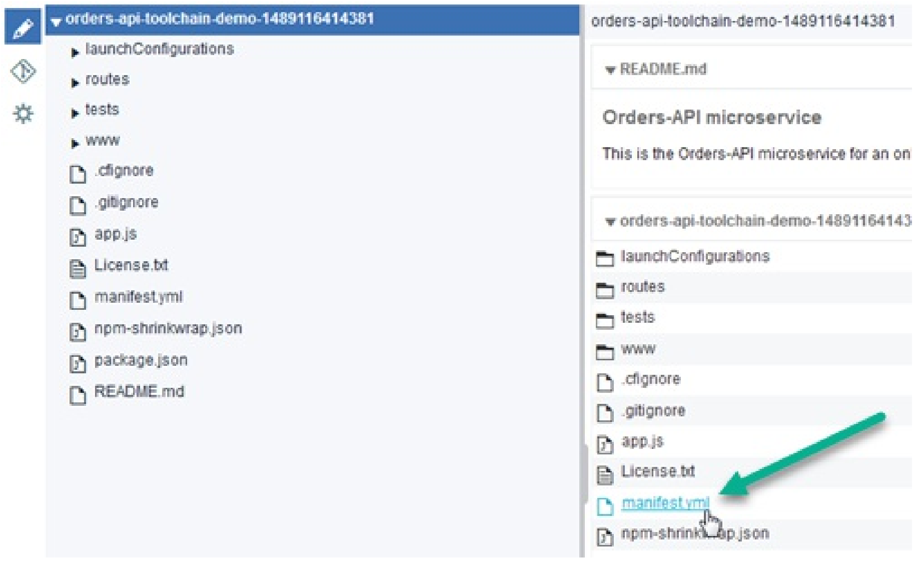
Name: JimP Test01
Title: What's your title?
Bio:
Time Zone: (UTC-05:00) Eastern Time (US & Canada) [account default]
Current Time: 12:32 PM (Your local time)
Phone: + Add Phone Number
SMS: + Add SMS Number

The list of countries PagerDuty supports is at: PagerDuty Country Support

Task 3: Break the application build

Let's see what happens when the application build is broken.

1. On the **simple-order-toolchain-*timestamp*** Toolchain page, click the **Eclipse Orion Web IDE** tile.
2. In the file directory, click manifest.yml to open the file.

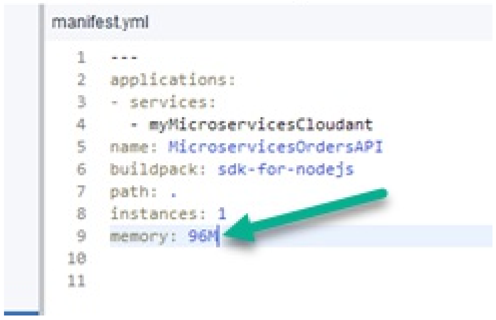


Orders-API microservice

This is the Orders-API microservice for an on

3. Update the value for memory to 96G. This setting intentionally

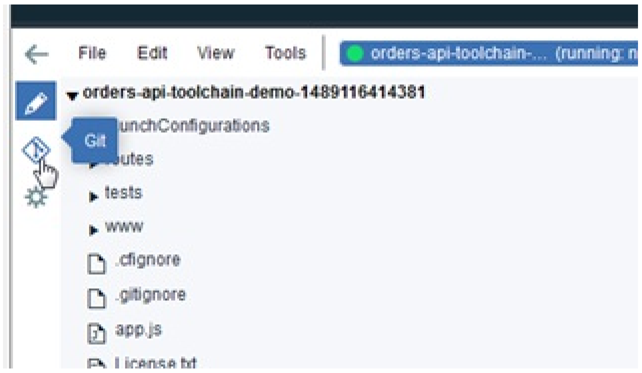
increases your memory to exceed the quota for your org. Your changes are automatically saved.



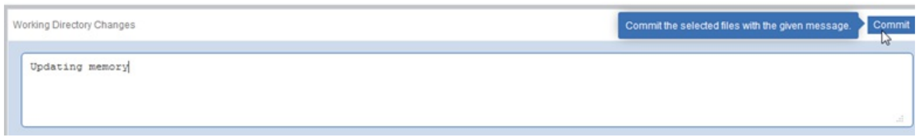
manifest.yml

```
---
applications:
- services:
  - myMicroservicesCloudant
name: MicroservicesOrdersAPI
buildpack: sdk-for-nodejs
path: .
instances: 1
memory: 96M
```

4. From the Eclipse Orion Web IDE menu, click the **Git** icon.

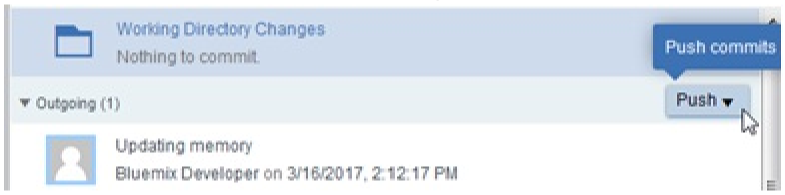


5. Enter a relevant comment and click **Commit** to push the changes in the local master branch.


Working Directory Changes
Commit the selected files with the given message.
Commit

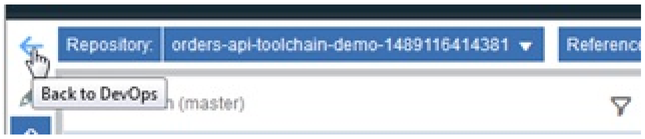
```
Updating memory
```

6. Click **Push** to put the changes into GitHub.

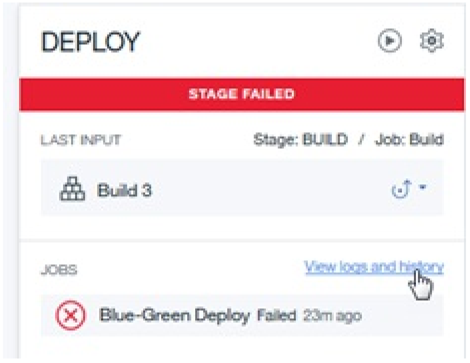


Updating memory
Bluemix Developer on 3/16/2017, 2:12:17 PM

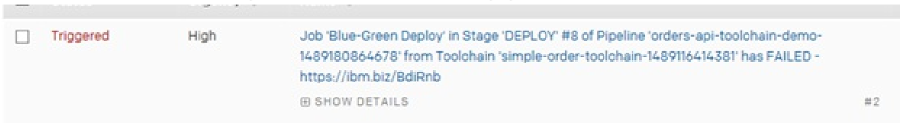
7. Your changes are automatically built and deployed in the pipeline.
8. Click the blue arrow in the upper left hand corner to return to the Toolchain.



9. Click the tile for the order-api-toolchain-lab Delivery Pipeline. Our change caused the deployment to fail.

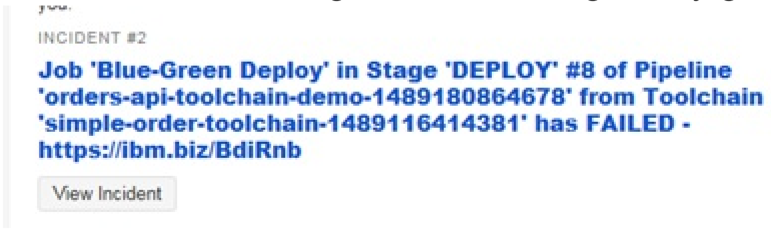


10. If you accepted the PagerDuty invitation and are logged into PagerDuty, the Incidents pages shows a triggered incident:



Triggered High Job 'Blue-Green Deploy' in Stage 'DEPLOY' #8 of Pipeline 'orders-api-toolchain-demo-1489180864678' from Toolchain 'simple-order-toolchain-1489116414381' has FAILED - https://ibm.biz/BdiRnb

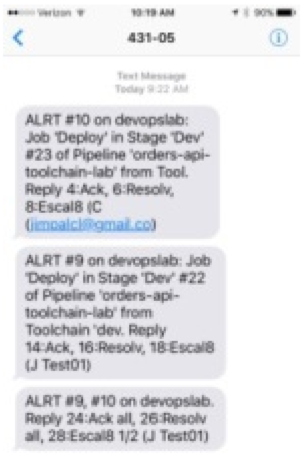
11. If you accepted the PagerDuty invitation and are logged into PagerDuty, the eMail account registered with PagerDuty get alerted.



INCIDENT #2

Job 'Blue-Green Deploy' in Stage 'DEPLOY' #8 of Pipeline 'orders-api-toolchain-demo-1489180864678' from Toolchain 'simple-order-toolchain-1489116414381' has FAILED - https://ibm.biz/BdiRnb

12. If you added a mobile phone number to your PagerDuty account, that phone will receive an alert and/or phone message.



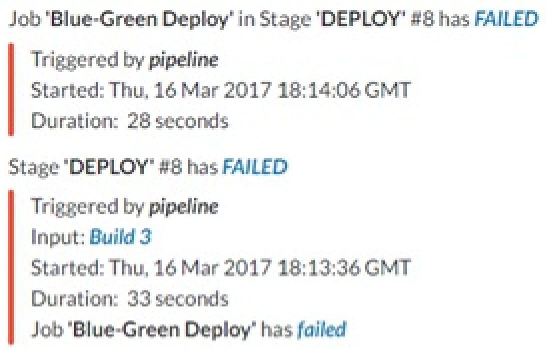
ALRT #10 on devopslab: Job 'Deploy' in Stage 'Dev' #23 of Pipeline 'orders-api-toolchain-lab' from Tool. Reply 4:Ack, 6:Resolv, 8:Escal8 (C (jimpalcl@gmail.co)

ALRT #9 on devopslab: Job 'Deploy' in Stage 'Dev' #22 of Pipeline 'orders-api-toolchain-lab' from Toolchain 'dev. Reply 14:Ack, 16:Resolv, 18:Escal8 (J Test01)

ALRT #9, #10 on devopslab. Reply 24:Ack all, 26:Resolv all, 28:Escal8 1/2 (J Test01)

13. Switch to the Slack browser tab, Slack also notifies the team about the

Deployment failure.



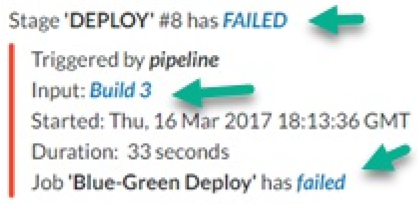
Job 'Blue-Green Deploy' in Stage 'DEPLOY' #8 has *FAILED*

Triggered by *pipeline*
Started: Thu, 16 Mar 2017 18:14:06 GMT
Duration: 28 seconds

Stage 'DEPLOY' #8 has *FAILED*

Triggered by *pipeline*
Input: *Build 3*
Started: Thu, 16 Mar 2017 18:13:36 GMT
Duration: 33 seconds
Job 'Blue-Green Deploy' has *failed*

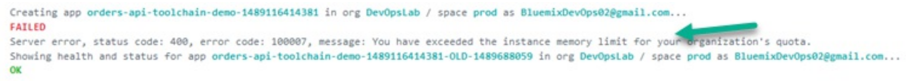
14. In Slack, there are a few links to go to the Stage, Job Input or Job. Click the **Failed** link for the Blue/Green Deploy job.



Stage 'DEPLOY' #8 has *FAILED*

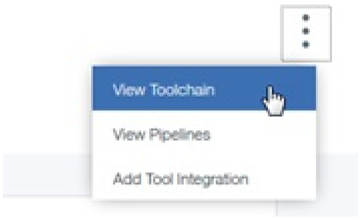
Triggered by *pipeline*
Input: *Build 3*
Started: Thu, 16 Mar 2017 18:13:36 GMT
Duration: 33 seconds
Job 'Blue-Green Deploy' has *failed*

15. Looking at the *Blue/Green Deploy* Job log (and scrolling down), we can see why it failed.



```
Creating app orders-api-toolchain-demo-1489116414381 in org DevOpsLab / space prod as BluemixDevOps02@gmail.com...
FAILED
Server error, status code: 400, error code: 100007, message: You have exceeded the instance memory limit for your organization's quota.
Showing health and status for app orders-api-toolchain-demo-1489116414381-OLD-1489688059 in org DevOpsLab / space prod as BluemixDevOps02@gmail.com...
OK
```

Return to the Toolchain by clicking on the vertical ellipsis on the upper right of the Bluemix console page and selecting **View Toolchain**.

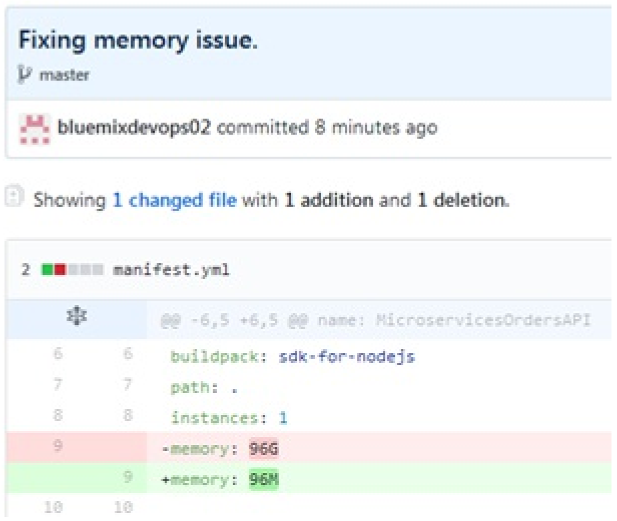


Task 4: Fix application

Now to fix the application.

1. On the **simple-order-toolchain-*timestamp*** Toolchain, click the **Eclipse Orion Web IDE** tile.
2. In the file directory, click manifest.yml to open the file.

3. Update the value for memory to 96m.
4. Now to Commit and Push the changes. From the Eclipse Orion Web IDE menu, click the **Git** icon.
5. Click **Commit** to put the changes in the local master branch (enter a relevant Commit comment).
6. Click **Push**. Your changes are automatically built and deployed in the pipeline.
7. Return to Toolchain page and click the Delivery Pipeline tile to watch the stages run in response to your code changes.
8. You could also display the Slack browser tab to display the Delivery Pipeline progress.
9. The deploy is successful and all the downstream stages run afterwards.
10. The Delivery Pipeline shows the Git URL and Git commit link. Click the Git commit link to show the changes that were committed that caused the Delivery Pipeline to execute.



Fixing memory issue.
master

bluemixdevops02 committed 8 minutes ago

Showing 1 changed file with 1 addition and 1 deletion.

2 manifest.yml

```
@@ -6,5 +6,5 @@ name: MicroservicesOrdersAPI
 buildpack: sdk-for-nodejs
 path: .
 instances: 1
-memory: 96G
+memory: 96M
```