

Lab 4: Add PagerDuty Integration

Objective

This lab show how to integrate your toolchain with PagerDuty so people get notified when things go wrong so problems can be fixed faster and reduce downtime.

Prerequisites

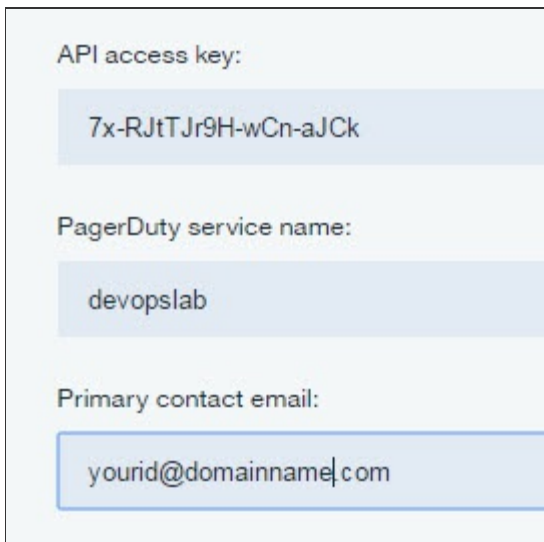
If you want to see email notifications from PagerDuty, you will need access to an eMail account. Once the integration is created, you can check that account to accept the invitation to join PagerDuty.

Tasks:

- [Task 1: Add PagerDuty to Toolchain](#)
- [Task 2: Verify PagerDuty works by breaking application build](#)
- [Task 3: Fix application](#)

Task 1: Add PagerDuty to Toolchain

1. On the devops-toolchain-lab toolchain's Tool Integrations page, click **Add a Tool**
2. Select **PagerDuty**
3. On the PagerDuty Configuration page:
 - Enter 7x-RJtTJr9H-wCn-aJcK as the API access key
 - Enter "devopslab" as the PagerDuty service name
 - Enter an email address you can access. Leave the phone number blank unless you want to receive text messages.
4. Click **Create Integration**



The screenshot shows a configuration form for PagerDuty integration. It has three input fields, each with a label and a light blue background for the text. The first field is labeled 'API access key:' and contains the text '7x-RJtTJr9H-wCn-aJcK'. The second field is labeled 'PagerDuty service name:' and contains the text 'devopslab'. The third field is labeled 'Primary contact email:' and contains the text 'yourid@domainname.com'.

5. You can access the eMail account to accept the PagerDuty invitation.

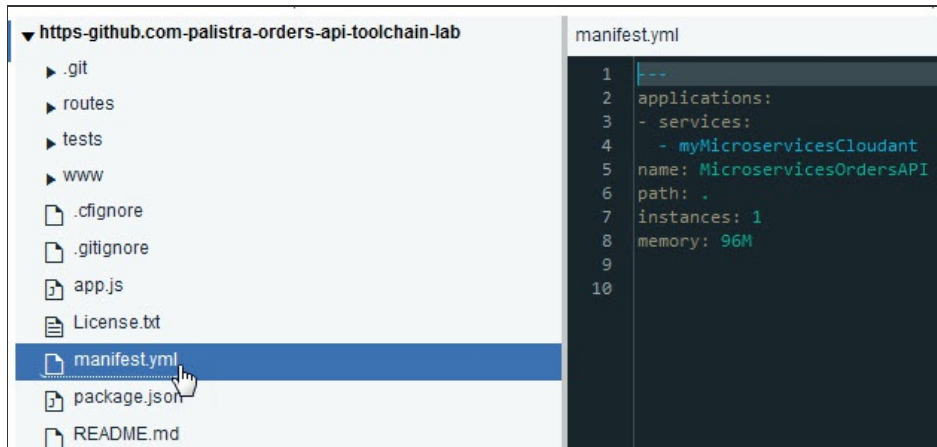
Task 2: Add Eclipse Orion Web IDE to Toolchain

We want to modify the application and one way is to use the Web IDE.

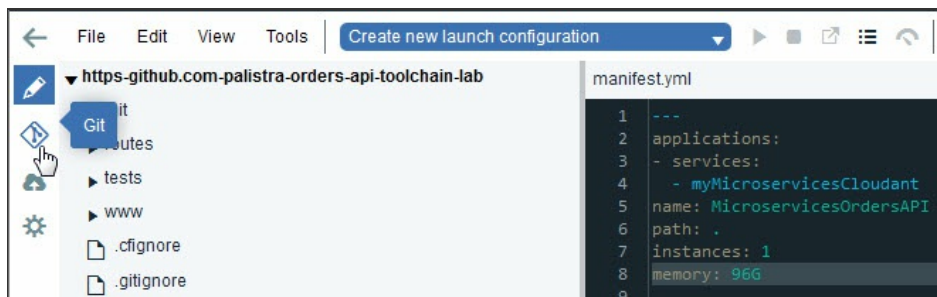
1. On the toolchain's Tool Integrations page, click the add button +
2. Select **Eclipse Orion Web IDE**.
3. No configuration is needed. so click **Create Integration**.

Task 3: Verify PagerDuty works by breaking application build

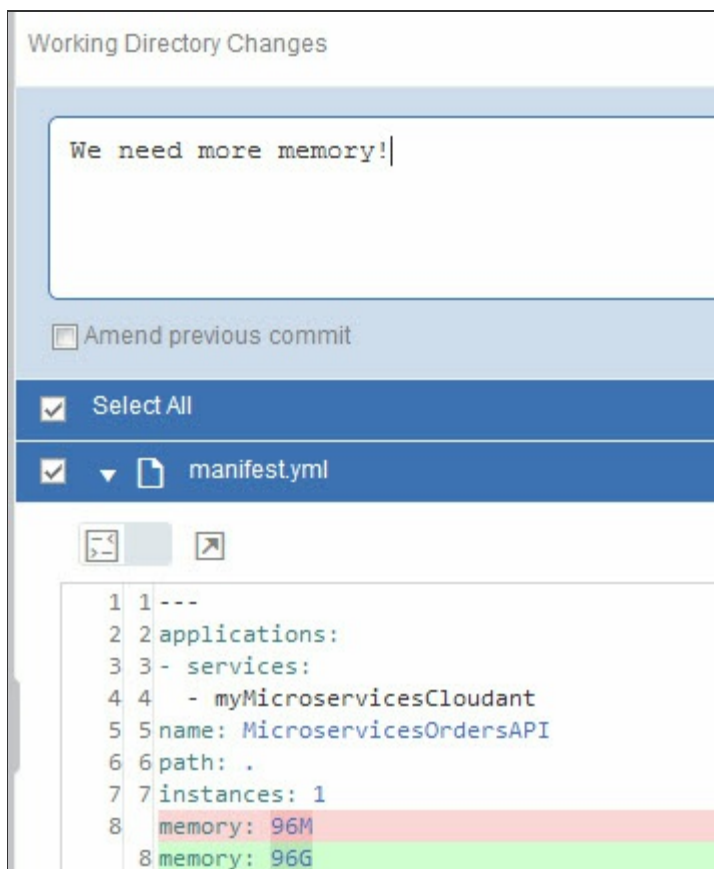
1. On the toolchain's Tool Integrations page, click the **Eclipse Orion Web IDE** tile. The GitHub repos are automatically loaded in your workspace. The Web IDE workspace is on the cloud.
2. In the file navigator, expand the orders-api-toolchain_name repo (if needed).
3. In the file directory, click manifest.yml to open the file.



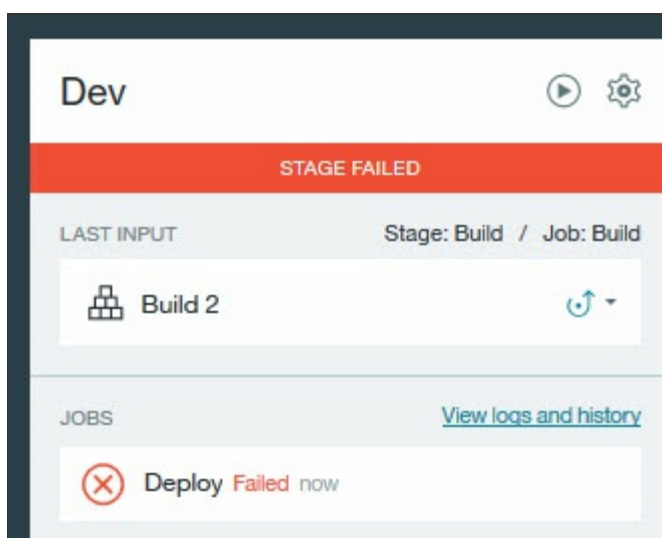
4. Update the value for memory to 96g. This setting intentionally increases your memory to exceed the quota for your org. Your changes are automatically saved.
5. Now to Push the changes. From the Eclipse Orion Web IDE menu, click the **Git** icon.



6. In the Working Directory Changes section, which is in the upper-right corner of the window, make sure that the changed file is selected. Enter a relevant comment.



7. Click **Commit** to put the changes in the local master branch.
8. Put these changes in the origin/master branch and click **Push**. Your changes are automatically built and deployed in the pipeline.
9. Return to devops-toolchain-lab toolchain's Tool Integrations page and click the pipeline tile for the orders-api microservice to watch the stages run in response to your commit (you may have to manually start the Build if it does not start automatically).
10. The Deploy fails.



11. The PagerDuty console (<https://devopslab01.pagerduty.com/incidents>) shows the incident:

	Urgency	#	Created On	Details	Service	Assigned To
	High	8	at 10:49 AM	Job 'Deploy' in Stage 'Dev' #4 of Pipeline 'orders-api-toolchain-lab' from Toolchain 'devops-toolchain-lab1' has FAILED - https://ibm.biz/BdsRnj	devopslab	Bluemix Cloud Developer

12. The account you

entered when you setup the PagerDuty integration will have an alert. The link in the email will allow you to view the incident on PagerDuty (assuming you accepted the invitation from PagerDuty).



Task 4: Fix application

Now to fix the application.

1. On the toolchain's Tool Integrations page, click the **Eclipse Orion Web IDE** tile.
2. In the file navigator, expand the orders-api-toolchain_name repo (if needed).
3. In the file directory, click manifest.yml to open the file.
4. Update the value for memory to 96m.
5. Now to Push the changes. From the Eclipse Orion Web IDE menu, click the **Git** icon.
6. In the Working Directory Changes section, which is in the upper-right corner of the window, make sure that the changed file is selected.
7. Click **Commit** to put the changes in the local master branch.
8. Put these changes in the origin/master branch and click **Push**. Your changes are automatically built and deployed in the pipeline.
9. Return to your toolchain's Tool Integrations page and click the pipeline tile for the orders-api microservice to watch the stages run in response to your commit (you may have to manually start the Build if it does not start automatically).
10. The deploy is successful. And all the downstream stages run afterwards.

