



Introduction to the Blue Planet Platform

PLF111ILT-A, Revision 1.0.1

Hands-on Training Lab Guide

LEGAL NOTICES

THIS DOCUMENT CONTAINS CONFIDENTIAL AND TRADE SECRET INFORMATION OF CIENA CORPORATION AND ITS RECEIPT OR POSSESSION DOES NOT CONVEY ANY RIGHTS TO REPRODUCE OR DISCLOSE ITS CONTENTS, OR TO MANUFACTURE, USE, OR SELL ANYTHING THAT IT MAY DESCRIBE. REPRODUCTION, DISCLOSURE, OR USE IN WHOLE OR IN PART WITHOUT THE SPECIFIC WRITTEN AUTHORIZATION OF CIENA CORPORATION IS STRICTLY FORBIDDEN.

EVERY EFFORT HAS BEEN MADE TO ENSURE THAT THE INFORMATION IN THIS DOCUMENT IS COMPLETE AND ACCURATE AT THE TIME OF PUBLISHING; HOWEVER, THE INFORMATION CONTAINED IN THIS DOCUMENT IS SUBJECT TO CHANGE.

While the information in this document is believed to be accurate and reliable, except as otherwise expressly agreed to in writing CIENA PROVIDES THIS DOCUMENT "AS IS" WITHOUT WARRANTY OR CONDITION OF ANY KIND, EITHER EXPRESS OR IMPLIED. The information and/or products described in this document are subject to change without notice. For the most up-to-date technical publications, visit www.ciena.com.

Copyright® 2023 Ciena® Corporation – All Rights Reserved

The material contained in this document is also protected by copyright laws of the United States of America and other countries. It may not be reproduced or distributed in any form by any means, altered in any fashion, or stored in a database or retrieval system, without the express written permission of Ciena Corporation.

Security

Ciena cannot be responsible for unauthorized use of equipment and will not make allowance or credit for unauthorized use or access.

Contacting Ciena

Corporate headquarters	410-694-5700 or 800-921-1144	www.ciena.com
Customer technical support/warranty		
In North America	1-800-CIENA24 (243-6224) 410-865-4961	Email: CIENA24@ciena.com
In Europe, Middle East, and Africa	800-CIENA-24-7 (800-2436-2247) +44-207-012-5508	Email: CIENA24@ciena.com
In Asia-Pacific	800-CIENA-24-7 (800-2436-2247) +81-3-6367-3989 +91-124-4340-600	Email: CIENA24@ciena.com
In Caribbean and Latin America	800-CIENA-24-7 (800-2436-2247) 410-865-4944 (USA)	Email: CIENA24@ciena.com
Sales and General Information	410-694-5700	E-mail: sales@ciena.com
In North America	410-694-5700 or 800-207-3714	E-mail: sales@ciena.com
In Europe	+44-207-012-5500 (UK)	E-mail: sales@ciena.com
In Asia	+81-3-3248-4680 (Japan)	E-mail: sales@ciena.com
In India	+91-124-434-0500	E-mail: sales@ciena.com
In Latin America	011-5255-1719-0220 (Mexico City)	E-mail: sales@ciena.com
Training		E-mail: learning@ciena.com

For additional office locations and phone numbers, please visit the Ciena website at www.ciena.com

Change History

Blue Planet Release	Revision	Publication Date	Reason for Change
	1.0	Mar 6, 2023	Initial release.
	1.0.1	Apr 7, 2023	Minor graphic update. No content update.

Contents

Lab 1: User Access Control (UAC).....	5
Lab 2: Platform Operations and Maintenance	29

Lab 1: User Access Control (UAC)

User Access control or UAC is a part of the Blue Planet Platform solution used for User and Tenant management. Under the scope of UAC, you will be able to create and modify Users and manage the Roles and Permissions assigned to users.

In this lab, you will learn how to create Tenants and Users, and how to assign permissions with pre-defined or custom Roles. You will also learn how to leverage REST API for Bulk user creation and how to use Swagger UI to develop REST API calls.

Objectives

- Log in to Blue Planet master Tenant
- Create a new Tenant
- Create a custom Role
- Log in to a customer Tenant
- Modify User information and change password
- Create a new User and assign Roles
- Use REST API and Swagger UI to create bulk users
- Verify User Privileges

Task 1: Log In to Blue Planet Master Tenant

The Blue Planet Platform uses Tenants to group users that are members of a common organization and have the same scope of responsibility in terms of managed resources. Thus, Tenants are used to separate the resource configuration scope for groups of users.

By default, the BPP system installation comes with one Master Tenant and one admin user preconfigured, and the admin user is assigned all default roles automatically.

Our Lab system has additional Tenants and users already configured.

In this lab, you will create one more Tenant for our *CustomerA* and you will create users for this Customer.

Since only a user with the *sysadmin* role in the master tenant can create new tenants, you must first log in to a **master** tenant as an **admin** user.

1. To log in to the Blue Planet Platform UI, from within your Student PC session, open a Chrome browser session (note the shortcut on the remote desktop screen) and enter <https://bp.lab.local> in the address space and press **Enter**, or click the **BP Application Bar** bookmark.

At the login screen, enter the *admin* user credentials and Tenant *master*, and click the **Login** button.

Username: **admin**

Password: **adminpw**

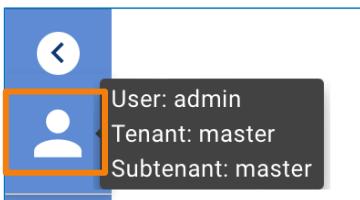
Tenant: **master**

The screenshot shows the login interface for the Blue Planet Platform. At the top center is a blue circle containing the letters 'bp'. Below it are three input fields: 'Username' (containing 'admin'), 'Password' (containing '.....'), and 'Tenant' (containing 'master'). At the bottom left is a link 'Forgot Password?' and at the bottom right is a large orange 'Log in' button.

2. In the next step you will be asked to enter the Subtenant. Leave this field blank and just click on the **Apply** button.

The screenshot shows a user interface with a blue header containing the 'bp' logo. Below the header is a form field labeled 'Subtenant:' with a dropdown arrow icon at the end. To the right of the dropdown is a blue rectangular button with the word 'Apply' in white. The entire form is enclosed in a light gray border.

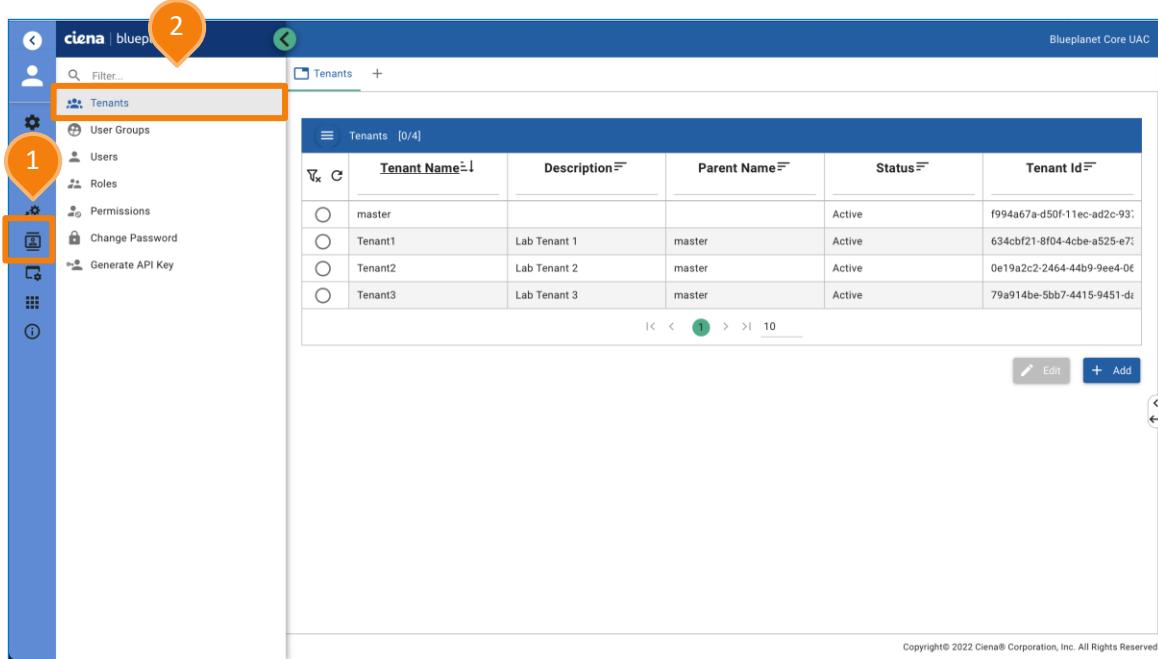
3. Once you are logged in, you can always verify your current user and tenant by hovering the mouse pointer over the User icon in the top left corner.



Task 2: Create a New Tenant for a Customer

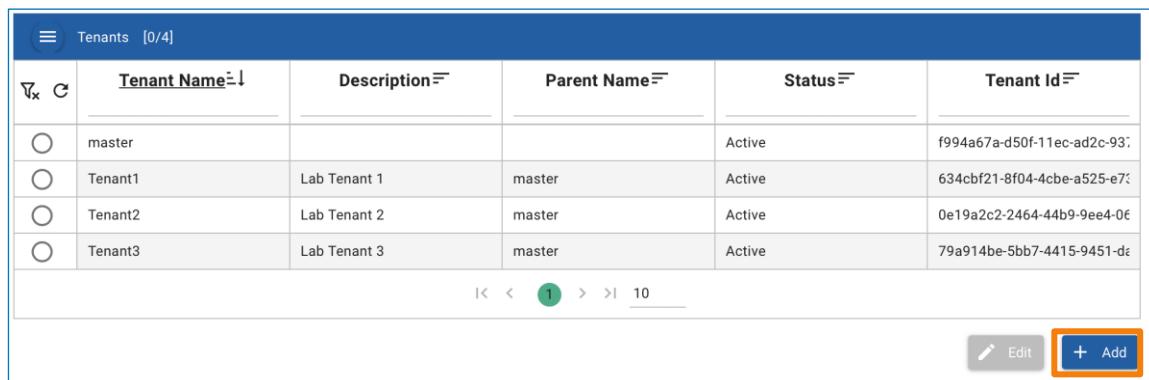
In this task, you will create a new Tenant for our CustomerA. That way, you can later separate resources and the users that will be available to that Customer.

1. To access the Tenant configuration, click on the **User Access Control** icon in the main menu and then click on the **Tenants** option. A new frame will open where you will be able to see the already configured Tenants.



	Tenant Name	Description	Parent Name	Status	Tenant Id
<input type="radio"/>	master			Active	f994a67a-d50f-11ec-ad2c-93
<input type="radio"/>	Tenant1	Lab Tenant 1	master	Active	634cbf21-8f04-4cbe-a525-e7
<input type="radio"/>	Tenant2	Lab Tenant 2	master	Active	0e19a2c2-2464-44b9-9ee4-0e
<input type="radio"/>	Tenant3	Lab Tenant 3	master	Active	79a914be-5bb7-4415-9451-d

2. Click the **+ Add** button to start creating a new Tenant.



	Tenant Name	Description	Parent Name	Status	Tenant Id
<input type="radio"/>	master			Active	f994a67a-d50f-11ec-ad2c-93
<input type="radio"/>	Tenant1	Lab Tenant 1	master	Active	634cbf21-8f04-4cbe-a525-e7
<input type="radio"/>	Tenant2	Lab Tenant 2	master	Active	0e19a2c2-2464-44b9-9ee4-0e
<input type="radio"/>	Tenant3	Lab Tenant 3	master	Active	79a914be-5bb7-4415-9451-d

3. A new frame will appear where you need to enter the new Tenant configuration. You will name this tenant **CustomerA** and add the description.
Select the **master** tenant as a parent from the drop-down menu and make sure to check the **Is Active** box so that the newly created Tenant will be immediately activated.
To finalize to process, click the **Apply** button.

The screenshot shows the 'Add Tenant' dialog box. It contains four fields: 'Tenant Name' (CustomerA), 'Description' (Customer's A Tenant), 'Parent' (master), and 'Is Active' (checked). The 'Is Active' field is highlighted with an orange border. Two orange numbered callouts point to the 'Apply' button (labeled '2') and the 'Is Active' checkbox (labeled '1').

* Tenant Name:	CustomerA
Description:	Customer's A Tenant
* Parent:	master
Is Active:	<input checked="" type="checkbox"/>

X Cancel Apply

4. You will be asked to confirm this operation with a password. Enter the admin password provided by the instructor and click the **Apply** button.

The screenshot shows the 'Confirm Password' dialog box. It has a single field labeled 'Password' containing several dots. Two orange numbered callouts point to the 'Password' field (labeled '1') and the 'Apply' button (labeled '2').

Please, enter password of the current logged in user

* Password:

X Cancel Apply

5. A green message box should appear in the top right corner showing that the new Tenant has been successfully created.
Verify that your Tenant is listed in the Tenants table and that it has *master* set as a parent.

The screenshot shows the 'Tenants' page in the Blue Planet Platform. On the left is a sidebar with icons for Tenants, User Groups, Users, Roles, Permissions, Change Password, and Generate API Key. The main area has tabs for 'Tenants' (selected), 'User Groups', and 'Tenants'. A green success message box at the top right says 'Tenant saved successfully.' An orange box highlights this message. Below it is a table with columns: Tenant Name, Description, Parent Name, Status, and Tenant Id. The table contains six rows: CustomerA (selected), master, Tenant1, Tenant2, and Tenant3. The 'CustomerA' row has a tooltip 'Customer's A Tenant' over the 'Description' field. The 'master' row is the parent of all other tenants. The 'Tenant1' through 'Tenant3' rows are lab tenants. At the bottom are 'Edit' and 'Add' buttons.

Tenant Name	Description	Parent Name	Status	Tenant Id
CustomerA	Customer's A Tenant	master	Active	0675567d-d3ac-4427-89c6-ff...
master			Active	f994a67a-d50f-11ec-ad2c-93...
Tenant1	Lab Tenant 1	master	Active	634cbf21-8f04-4cbe-a525-e7...
Tenant2	Lab Tenant 2	master	Active	0e19a2c2-2464-44b9-9ee4-0e...
Tenant3	Lab Tenant 3	master	Active	79a914be-5bb7-4415-9451-dc...

NOTE: When you create a new Tenant, an *admin* user for this Tenant will also be created automatically. You will set up a new password for this *admin* user when you log in to CustomerA Tenant in the next tasks.

Task 3: Create a Custom Role

For this customer, you plan to create several users with different Roles and Permissions.

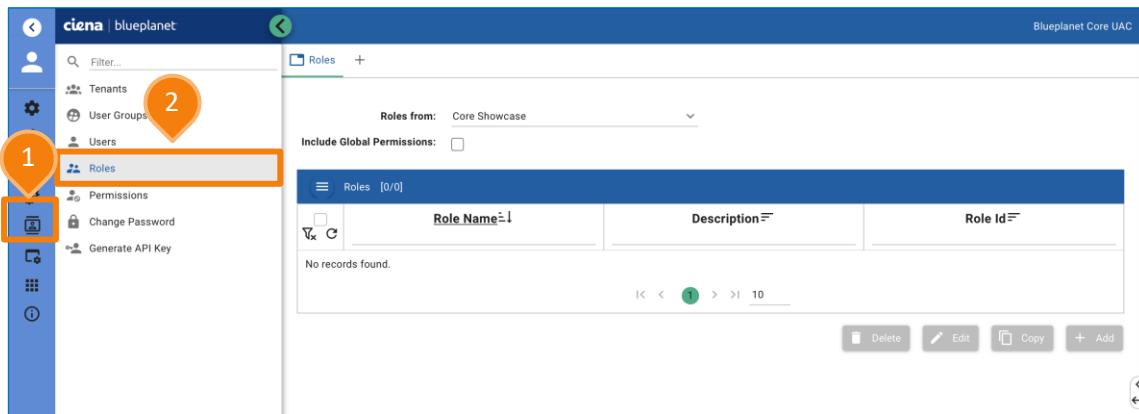
Each user in Blue Planet is assigned one or more Roles and each Role has a set of specific permissions which control specific administrative and resource management actions that a user can perform.

There are default roles provided as part of the BPP system to assist you with establishing main user roles, but you can also create Custom Roles.

You will now create one new custom Role while you are still logged in as a master tenant.

To create a custom Role, you will copy one of the default Roles and modify the permissions assigned to it.

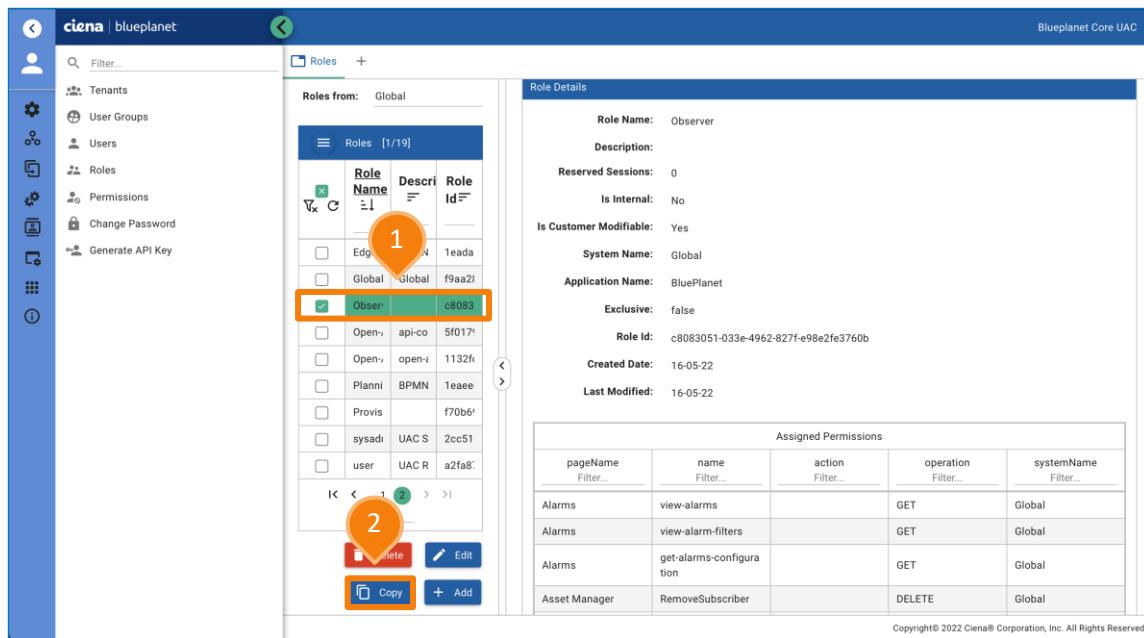
1. To access the Roles configuration, click on the **UAC** icon on the menu and then click on the **Roles** option.



2. From the **Roles from** dropdown menu, choose **Global**. This will make all the pre-defined roles show up in the table. Click on the **Page 2** button below the table.

<input type="checkbox"/>	Role Name	Description	Role Id
<input type="checkbox"/>	admin	UAC Administrator	7571162a-61a8-47f1-92da-5bf31d794b2f
<input type="checkbox"/>	Administrators	Administrators Role	1e4f3ed0-d510-11ec-8b34-bf0c8588e67d
<input type="checkbox"/>	Application admin		a84009d9-bb35-41c4-92db-06f5121fe43b
<input type="checkbox"/>	bpmnAdmin	BPMN Administrator Role	1e4ce27a-d510-11ec-8b34-0fb654ec9298
<input type="checkbox"/>	bpmnDeveloper	BPMN Developer Role	1e5078a4-d510-11ec-8b34-47c36dff22f7
<input type="checkbox"/>	bpmnManager	BPMN Manager Role	1e4c10ac-d510-11ec-8b34-1beafcd4b31
<input type="checkbox"/>	bpmnMDSO	Workflow MDSO Role	1e4e3940-d510-11ec-8b34-1f5233d505aa
<input type="checkbox"/>	bpmnSOO	Workflow SOO Role	1e4d75b4-d510-11ec-8b34-2f60850358a
<input type="checkbox"/>	bpmnUser	BPMN User Role	1e4ae894-d510-11ec-8b34-eb8ed332f457
<input type="checkbox"/>	Core	BPMN Role for assigned groups	1eac03d6-d510-11ec-a32b-9f1fff64e8da

3. Select the Role named **Observer** by checking the box in the first column. The details for this Role will be shown in a new frame. Once selected, click the **Copy** button.



The screenshot shows the Blue Planet Core UAC interface. On the left, there is a sidebar with various navigation options: Tenants, User Groups, Users, Roles, Permissions, Change Password, and Generate API Key. The main area has a title bar "ciena | blueplanet" and a search bar "Filter...". Below the search bar is a "Roles from: Global" dropdown. The central part of the screen displays a table titled "Roles [1/19]" with columns: Role Name, Description, and Role Id. One row in the table is highlighted with a green background and contains the role name "Observer". A large orange circle with the number "1" is drawn around this row. At the bottom of the table, there are buttons for "Delete", "Edit", and "Copy". Another orange circle with the number "2" is drawn around the "Copy" button. To the right of the table, a modal window titled "Role Details" is open. It contains the following information:

Role Details	
Role Name:	Observer
Description:	
Reserved Sessions:	0
Is Internal:	No
Is Customer Modifiable:	Yes
System Name:	Global
Application Name:	BluePlanet
Exclusive:	false
Role Id:	c8083051-033e-4962-827f-e98e2fe3760b
Created Date:	16-05-22
Last Modified:	16-05-22

Below the "Role Details" section is another table titled "Assigned Permissions" with columns: pageName, name, action, operation, and systemName. The data in this table is as follows:

pageName	name	action	operation	systemName
Alarms	view-alarms	GET	Global	
Alarms	view-alarm-filters	GET	Global	
Alarms	get-alarms-configuration	GET	Global	
Asset Manager	RemoveSubscriber		DELETE	Global

At the bottom right of the "Role Details" dialog, it says "Copyright © 2022 Ciena® Corporation, Inc. All Rights Reserved."

4. A new Role dialogue will open in the right frame with copied permissions from the Observer Role. You want to give it a new name and change the assigned permissions. Enter the following:

Role Name: Custom-Observer
Description: Custom Observer Role for Customer A
Exclusive: leave **unchecked**

NOTE: When the role is an Exclusive, it cannot be combined with any other role while assigning it to a user or user group.

Next enter the text **delete** in the operation column filter from the Assigned Permissions table to filter out all permissions to delete resources.

Add Role

1

* Role Name:	Custom-Observer
* Description:	Custom Observer Role for Customer A
Exclusive:	<input type="checkbox"/>

Available Permissions				
pageName Filter...	name Filter...	action Filter...	operation Filter...	systemName Filter...
Create Saved Layout	Create	GET	Global	
Delete Saved Layout	Delete	GET	Global	
Set Default Saved Layout	Modify	GET	Global	
Add to Scratch Pad	Launch Add to Scratch Pad	Launch	GET	Global
Alarms	delete-a-larm-filter		DELETE	Global

2

Assigned Permissions				
pageName Filter...	name Filter...	action Filter...	operation Filter...	systemName Filter...
Asset Manager	RemoveSubscriber		DELETE	Global
Market	DeleteResourceHistory		DELETE	Global
Security	delete-api-keys		DELETE	Global

5. Select all assigned permissions that have **operation=delete** with **Shift or Ctrl + click**. The chosen permission will be marked with a green color.
Once the permissions are chosen, remove them from the **Assigned Permissions** table by clicking the < button.

NOTE: You will need to scroll down the frame to find the < button.

The screenshot shows a table titled "Assigned Permissions" with five columns: pageName, name, operation, and systemName. The first column has filters for "pageName", "name", and "operation". The second column has filters for "name" and "operation". The third column has a filter for "operation". The fourth column has a filter for "systemName". There are three rows of data:

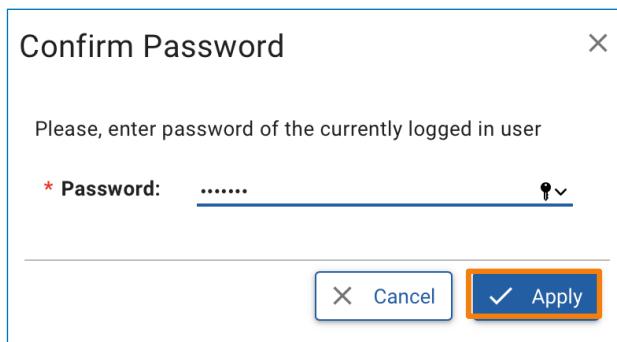
pageName	name	operation	systemName
Asset Manager	Remove eSubscriber	DELETE	Global
Market	DeleteResource History	DELETE	Global
Security	delete-a-pi-keys	DELETE	Global

Callout 1 points to the "operation" column header. Callout 2 points to the "Remove" button in the first row.

6. Scroll down to the end of the Permissions frame and click the **Apply** button.

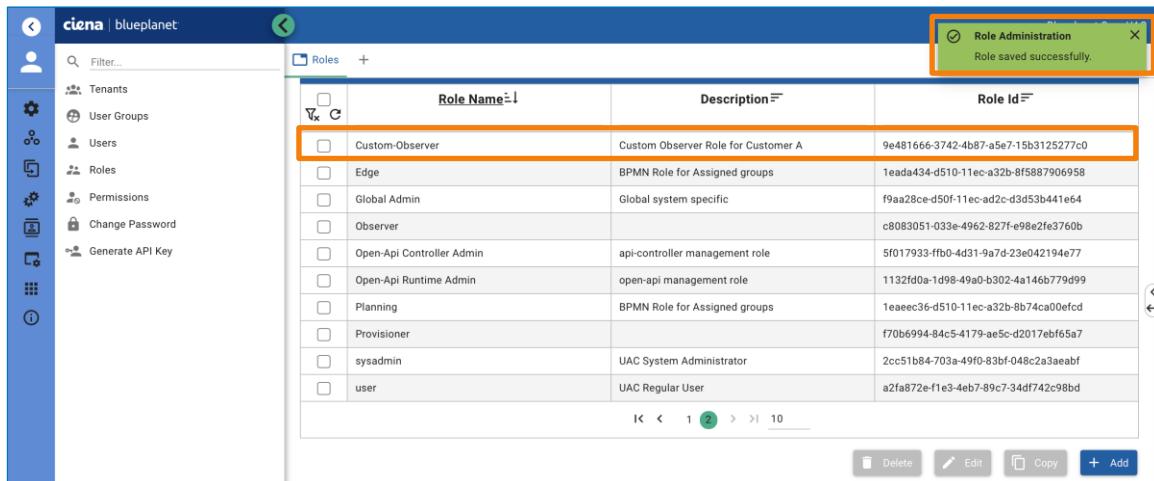


7. You will be asked to confirm this operation with a password. Enter the admin password provided by the instructor and click the **Apply** button.



8. A green message box should appear in the top-right corner showing that the new Role has been successfully created.

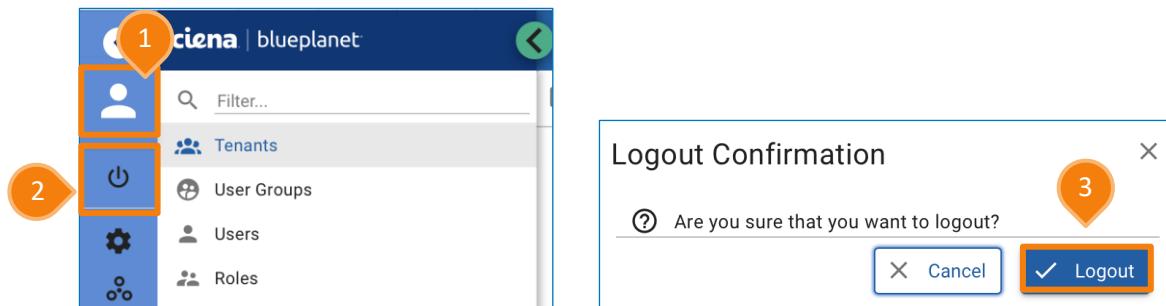
Verify that your Role is listed in the Roles table.



The screenshot shows the 'Roles' page in the Blue Planet Platform. On the left is a sidebar with icons for Tenants, User Groups, Users, Roles, Permissions, Change Password, and Generate API Key. The main area has a search bar and a table titled 'Roles'. The table has columns for 'Role Name', 'Description', and 'Role Id'. A new row is highlighted with an orange border, representing the 'Custom-Observer' role. At the top right of the table area, there is a green message box with a checkmark icon and the text 'Role saved successfully.'.

Role Name	Description	Role Id
Custom-Observer	Custom Observer Role for Customer A	9e481666-3742-4b87-a5e7-15b3125277c0
Edge	BPMN Role for Assigned groups	1ead434-d510-11ec-a32b-8f5887906958
Global Admin	Global system specific	f9aa28ce-d50f-11ec-ad2c-d3d53b441e64
Observer		c8083051-033e-4962-827f-e98e2fe3760b
Open-Api Controller Admin	api-controller management role	5f017933-ffb0-4d31-9a7d-23e04219e77
Open-Api Runtime Admin	open-api management role	1132fd0a-1d98-49a0-b302-4a146b779a99
Planning	BPMN Role for Assigned groups	1eaec36-d510-11ec-a32b-8b74ca00efcd
Provisioner		f70b6994-84c5-4179-ae5c-d2017ebf65a7
sysadmin	UAC System Administrator	2cc51b84-703a-49f0-83bf-048c2a3eeabf
user	UAC Regular User	a2fa872e-f1e3-4eb7-89c7-34df742c98bd

9. As a final step of this task, you will log out of the Blue Planet system.
 Click the **User** icon in the top left corner and the logout icon will show up. Then click the **Logout** icon and when the Logout confirmation box shows up confirm by clicking the **Logout** button.



Task 4: Log In to a Customer Tenant

In Task 2, you have created a specific Tenant for our CustomerA.

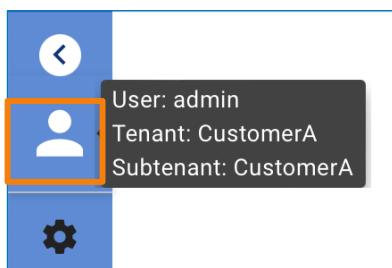
You will now log in as an administrator into the CustomerA tenant, so when you create new Users, they will automatically become members of the CustomerA tenant.

1. At the login screen, enter the **admin** user credentials and Tenant **CustomerA** and click the **Login** button.

Username: **admin**
Password: **adminpw**
Tenant: **CustomerA**

The image shows a login interface with a blue header containing the 'bp' logo. Below it are three input fields: 'Username' (containing 'admin'), 'Password' (containing redacted dots), and 'Tenant' (containing 'CustomerA'). At the bottom are two buttons: 'Forgot Password?' and a large orange-bordered 'Log in' button.

2. Verify your user and tenant by hovering the mouse pointer over the User icon in the top left corner.

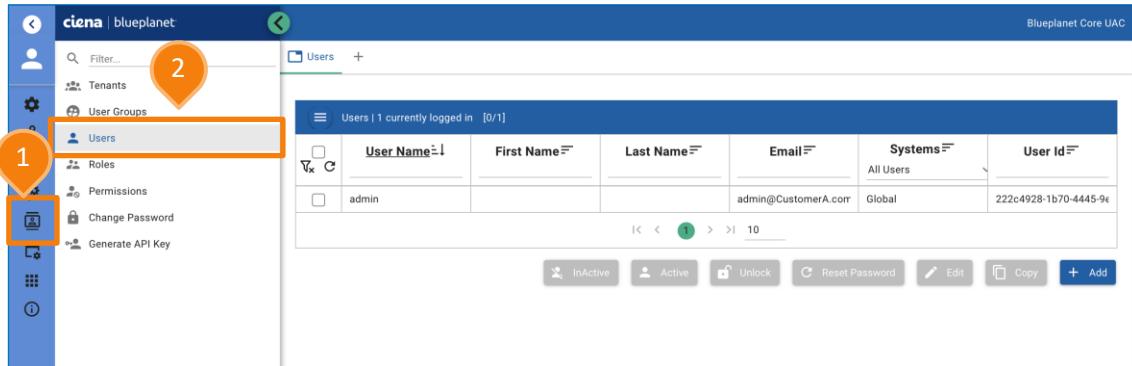


Task 5: Create and Modify Users in CustomerA Tenant

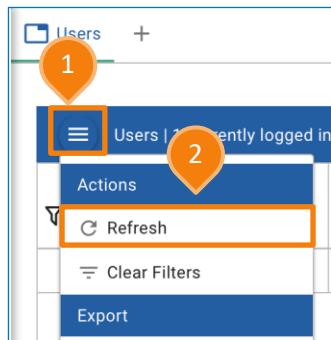
You are now logged in as admin in the CustomerA tenant, so all User configuration is applicable for this Tenant only. When you create new Users, they will become members of the CustomerA tenant.

But first, you need to change the credentials and update the information for the CustomerA admin that was previously created automatically when you created this Tenant.

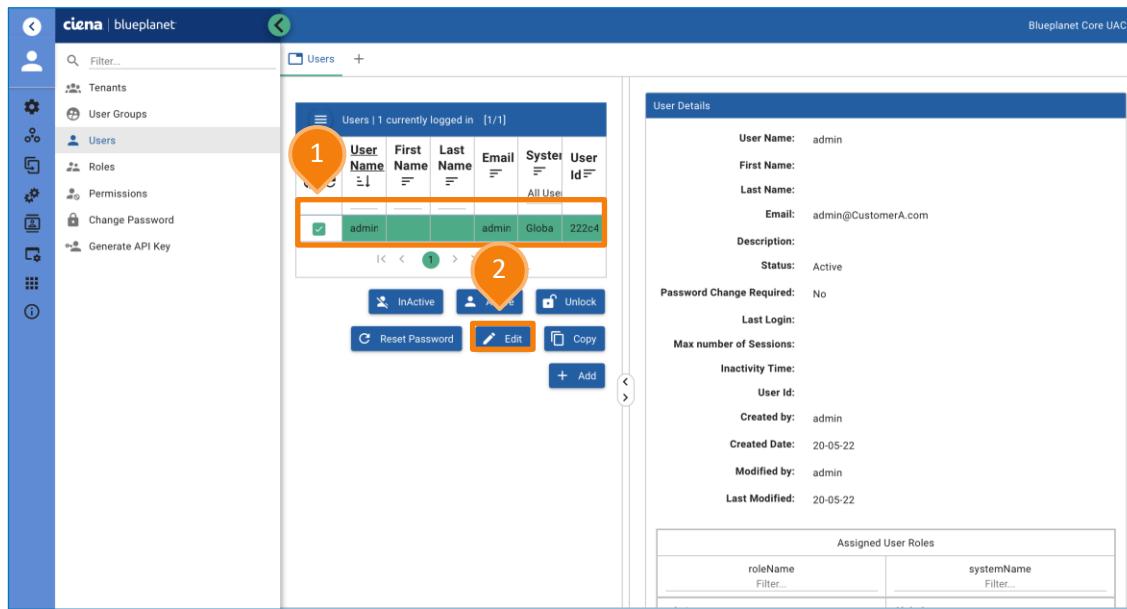
1. Click the **UAC** icon from the main menu and then click the **Users** option to start the User configuration.



NOTE: If multiple users are shown, you might need to refresh the list by clicking the **Menu** icon in the **Users** table and then clicking the **Refresh** option.

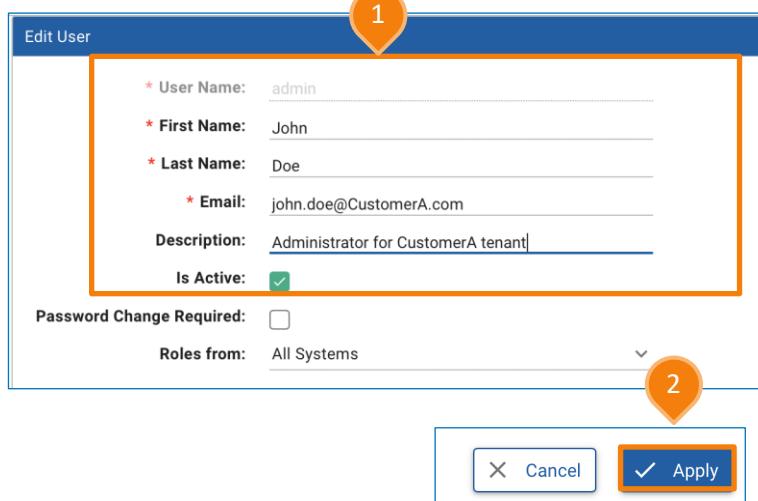


2. Select the user **admin** and click the **Edit** button to modify user information.



The screenshot shows the 'Users' section of the Blue Planet Core UAC. On the left sidebar, 'Users' is selected. The main area displays a table of users with one row selected for the user 'admin'. An orange circle labeled '1' highlights the 'Edit' button at the bottom of the table. Another orange circle labeled '2' highlights the 'User Details' panel on the right side of the screen, which contains fields for User Name, First Name, Last Name, Email, and other user metadata.

3. Enter the **Name**, **email**, and **description**, and make sure that the **Is Active** box is checked. Once you enter the required information, click the **Apply** button.



The screenshot shows the 'Edit User' dialog box. It contains fields for User Name (admin), First Name (John), Last Name (Doe), Email (john.doe@CustomerA.com), and Description (Administrator for CustomerA tenant). The 'Is Active' checkbox is checked. Below the form, there are buttons for 'Cancel' and 'Apply'. An orange circle labeled '1' points to the 'Edit User' title bar, and another orange circle labeled '2' points to the 'Apply' button.

4. You will need to confirm this with a password. Enter the admin password and click **Apply**. The user should be successfully saved, and the green notification box should appear as a confirmation.

Confirm Password

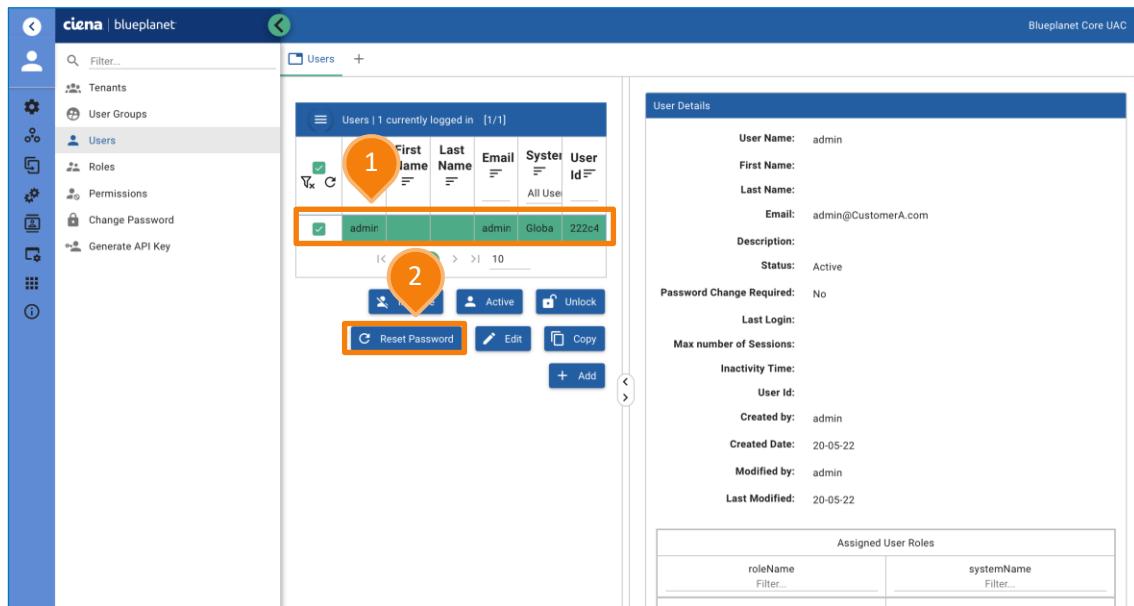
Please, enter password of the currently logged in user

* Password: 1

User Administration 2

User saved successfully.

- Next, you will change the password for the CustomerA admin user. Choose the user in the table and click the **Reset Password** button.



The screenshot shows the 'User Details' page for the 'admin' user. The 'User Details' section on the right displays the following information:

- User Name: admin
- First Name:
- Last Name:
- Email: admin@CustomerA.com
- Description:
- Status: Active
- Password Change Required: No
- Last Login:
- Max number of Sessions:
- Inactivity Time:
- User Id:
- Created by: admin
- Created Date: 20-05-22
- Modified by: admin
- Last Modified: 20-05-22

The 'Assigned User Roles' section below is empty.

- Enter the old and the new password and click **Apply**.

Old Password: **adminpw**

New Password: **12345678**

User Details

* Old Password: 1

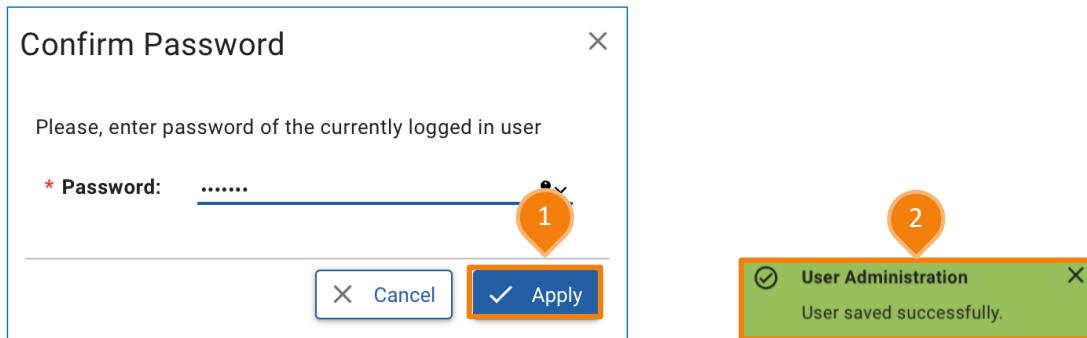
* New Password: 1

* Confirm Password: 1

User Administration 2

User saved successfully.

7. You will need to confirm this operation with a password. Enter the **old admin password** and click **Apply**. The user should be successfully saved, and the green notification box should appear to confirm that.



8. Next, you will create one additional user in the CustomerA tenant. This will be a user with Provisioning Role. To start the process, click the **+ Add** button from the Users screen.



9. Enter the user information as shown in the image below and move the **Provisioning** Role from the **Available Roles** to **Assigned User Roles**. Make sure that the **Is Active** box is checked. Click the **Apply** button.

Username: **prov1**
 Password: **12345678**

Add User

* User Name:	prov1																
* First Name:	Jane																
* Last Name:	Doe																
* Email:	jane.doe@customerA.com																
Description:	Provisioner for Customer A																
Password: 																
Confirm Password: 																
Is Active:	<input checked="" type="checkbox"/>																
Password Change Required:	<input type="checkbox"/>																
Roles from:	All Systems																
<table border="1"> <thead> <tr> <th colspan="2">Available Roles</th> </tr> <tr> <th>roleName</th> <th>systemName</th> </tr> </thead> <tbody> <tr> <td>admin</td> <td>Global</td> </tr> <tr> <td>Administrators</td> <td>Global</td> </tr> <tr> <td>Application admin</td> <td>Global</td> </tr> </tbody> </table> <table border="1"> <thead> <tr> <th colspan="2">Assigned User Roles</th> </tr> <tr> <th>roleName</th> <th>systemName</th> </tr> </thead> <tbody> <tr style="background-color: #00c080; color: white;"> <td>Provisioner</td> <td>Global</td> </tr> </tbody> </table>		Available Roles		roleName	systemName	admin	Global	Administrators	Global	Application admin	Global	Assigned User Roles		roleName	systemName	Provisioner	Global
Available Roles																	
roleName	systemName																
admin	Global																
Administrators	Global																
Application admin	Global																
Assigned User Roles																	
roleName	systemName																
Provisioner	Global																
<input type="button" value="Cancel"/> <input checked="" type="button" value="Apply"/>																	

10. You will need to confirm with a password. Please enter the **old admin password** and click **Apply**. The user should be successfully saved, and the green notification box should appear as confirmation.

Confirm Password

Please, enter password of the currently logged in user

* Password: 

 User Administration X

User saved successfully.

11. Your Users table should now have two users configured and it should look like this.

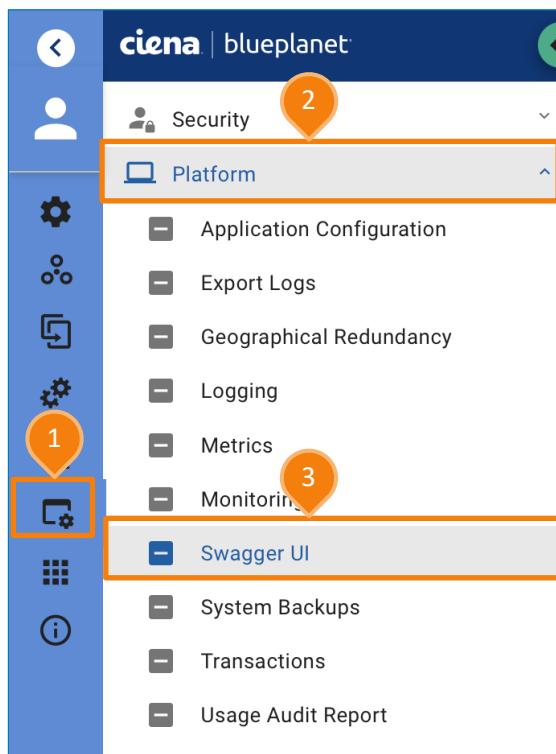
Users 1 currently logged in [0/2]						
<input type="checkbox"/>	User Name	First Name	Last Name	Email	Systems	User Id
<input type="checkbox"/>	admin	John	Doe	john.doe@CustomerA.co	Global	222c4928-1b70-4445-9e
<input type="checkbox"/>	prov1	Jane	Doe	jane.doe@customerA.cc	Global	a3b3f969-0272-43a9-8a

Task 6: Use REST API and Swagger UI to Create Bulk Users

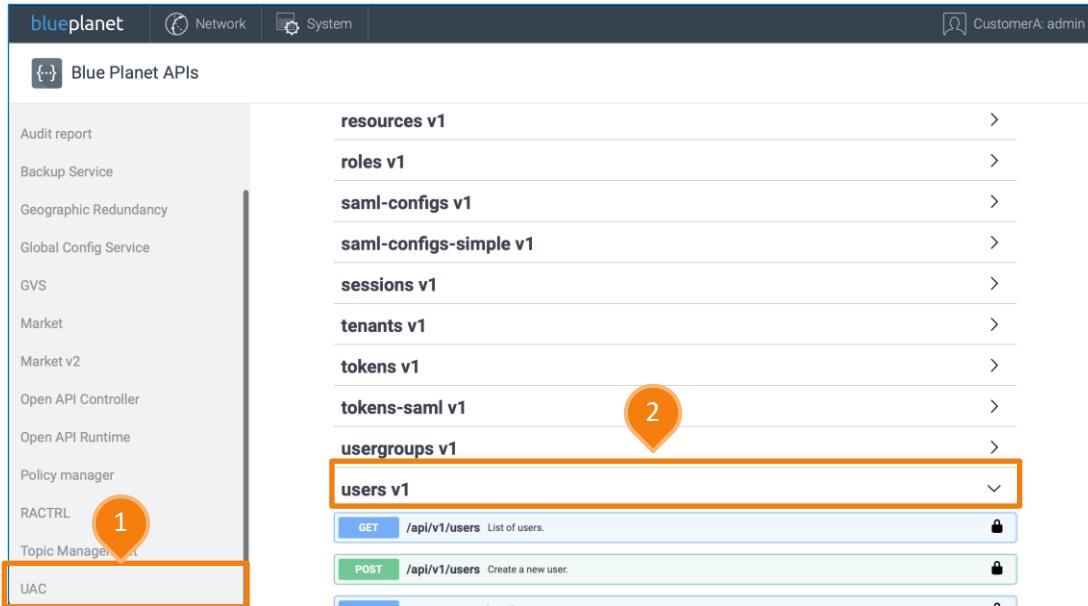
In this task, you will learn to develop and use REST API calls to create bulk sets of users. The plan is to create two new users in the CustomerA tenant and assign the previously created custom Role named “Custom-Observer”.

To create and execute REST API calls, you will use Swagger UI. Swagger is a suite of API developer tools and it is accessible from the Blue Planet UI.

1. To open Swagger UI, click the **System** icon from the menu, choose **Platform**, and then click the **Swagger UI** option. Swagger UI will open in a new browser tab.

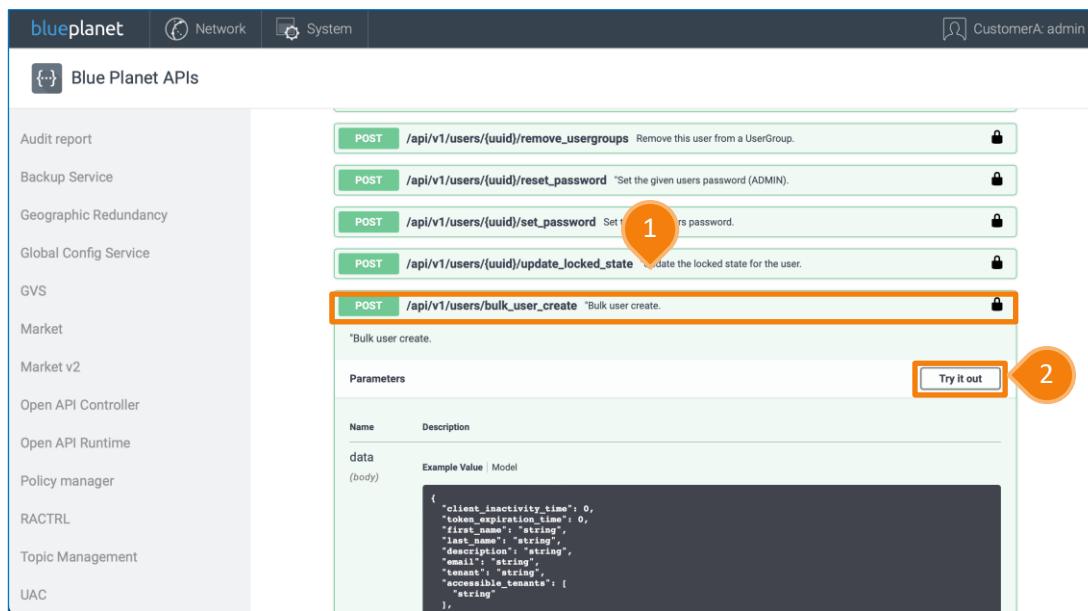


2. In the Swagger UI, click the **UAC** option from the left side menu. Scroll down and click on the **users v1** row.



The screenshot shows the Blue Planet APIs Swagger UI. On the left sidebar, the 'UAC' option is highlighted with a red box and a circled '1'. On the main panel, the 'users v1' row under the 'resources v1' section is highlighted with a red box and a circled '2'. Below it, there are two buttons: 'GET /api/v1/users' (List of users) and 'POST /api/v1/users' (Create a new user).

3. Click on the **/api/v1/users/bulk_users_create** to open this POST call. Click the **Try it out** button to start creating the API call.



The screenshot shows the same Swagger UI interface. The 'bulk_user_create' POST call under the 'POST /api/v1/users/bulk_user_create' section is highlighted with a red box and a circled '1'. To its right, the 'Try it out' button is also highlighted with a red box and a circled '2'.

4. You will create 2 users: **observer1** and **observer2**. Both users will be added to the **CustomerA** tenant and assigned the previously created “**Custom-Observer**” Role and a password: **12345678**. Copy the following code in the **Edit Value** frame and click the **Execute** button:

```
[{
    "first_name": "First",
```

```

    "last_name": "Observer",
    "description": "Read only observer for CustomerA",
    "email": "observer.one@customerA.com",
    "tenant": "CustomerA",
    "roles": [{"name": "Custom-Observer",
"application": "BluePlanet"}],
    "password": "12345678",
    "password_change_required": true,
    "username": "observer1",
    "is_active": true
}, {
    "first_name": "Second",
    "last_name": "Observer",
    "description": "Read only observer for CustomerA",
    "email": "observer.two@customerA.com",
    "tenant": "CustomerA",
    "roles": [{"name": "Custom-Observer",
"application": "BluePlanet"}],
    "password": "12345678",
    "password_change_required": true,
    "username": "observer2",
    "is_active": true
}
]

```

Parameters

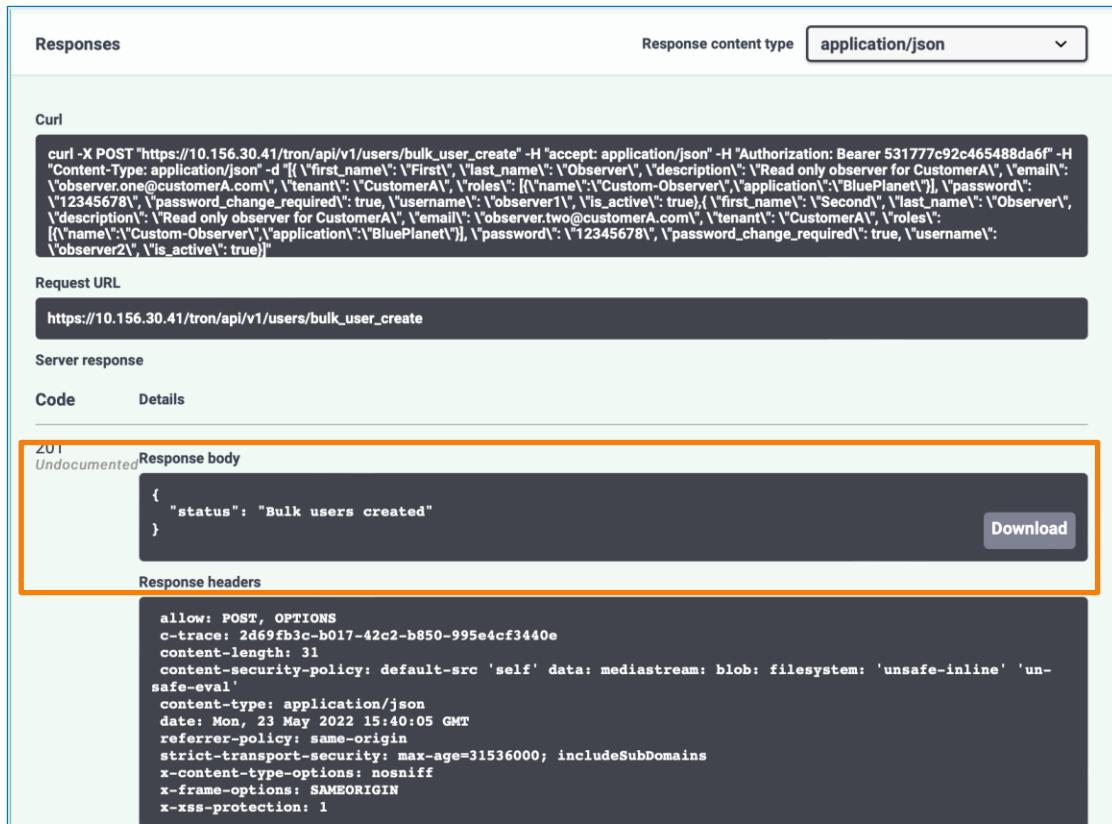
Name **Description**

Name	Description
data (body)	<input type="button" value="Edit Value Model"/> 1 <pre>{ "first_name": "First", "last_name": "Observer", "description": "Read only observer for CustomerA", "email": "observer.one@customerA.com", "tenant": "CustomerA", "roles": ["Custom-Observer"], "password": "12345678", "password_change_required": true, "username": "observer1", "is_active": true }, { "first_name": "Second", "last_name": "Observer", "description": "Read only observer for CustomerA", "email": "observer.two@customerA.com", "tenant": "CustomerA", "roles": ["Custom-Observer"], "password": "12345678", "password_change_required": true, "username": "observer2" }]</pre> <input type="button" value="Cancel"/>

Parameter content type

2

5. Check the Response body and verify that the status returned has code **201** and the message “**Bulk users created**”.



```

Responses
Response content type application/json

Curl
curl -X POST "https://10.156.30.41/tron/api/v1/users/bulk_user_create" -H "accept: application/json" -H "Authorization: Bearer 531777c92c465488da6f" -H "Content-Type: application/json" -d "[{"first_name": "First", "last_name": "Observer", "description": "Read only observer for CustomerA", "email": "Observer.one@customerA.com", "tenant": "CustomerA", "roles": [{"name": "Custom-Observer", "application": "BluePlanet"}, {"password": "12345678", "password_change_required": true, "username": "Observer1", "is_active": true}, {"name": "Second", "last_name": "Observer", "description": "Read only observer for CustomerA", "email": "Observer.two@customerA.com", "tenant": "CustomerA", "roles": [{"name": "Custom-Observer", "application": "BluePlanet"}, {"password": "12345678", "password_change_required": true, "username": "Observer2"}, {"is_active": true}]}]"

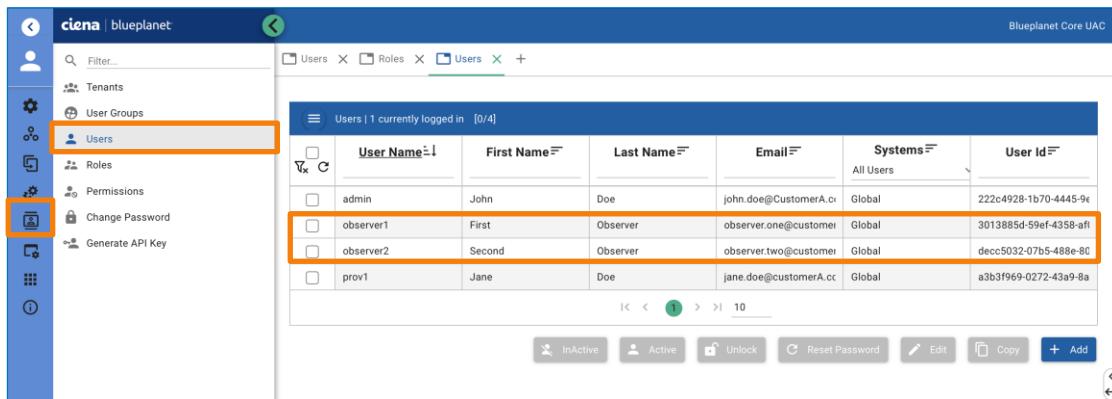
Request URL
https://10.156.30.41/tron/api/v1/users/bulk_user_create

Server response
Code Details
201 Undocumented Response body
{
  "status": "Bulk users created"
}
Download

Response headers
allow: POST, OPTIONS
c-trace: 2d69fb3c-b017-42c2-b850-995e4cf3440e
content-length: 31
content-security-policy: default-src 'self' data: mediastream: blob: filesystem: 'unsafe-inline' 'unsafe-eval'
content-type: application/json
date: Mon, 23 May 2022 15:40:05 GMT
referrer-policy: same-origin
strict-transport-security: max-age=31536000; includeSubDomains
x-content-type-options: nosniff
x-frame-options: SAMEORIGIN
x-xss-protection: 1

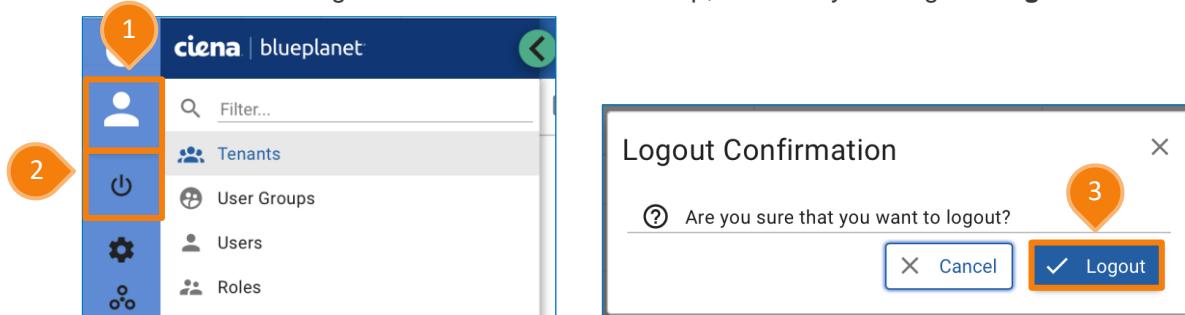
```

6. Go back to the Blue Planet UI tab and verify that the users are created on the **UAC > Users** page.



User Name	First Name	Last Name	Email	Systems	User Id
admin	John	Doe	john.doe@customerA.cc	Global	22c4928-1b70-4445-9e
observer1	First	Observer	observer.one@customerA.com	Global	3013885d-59ef-435b-aef
observer2	Second	Observer	observer.two@customerA.com	Global	decc5032-07b5-488e-8c
prov1	Jane	Doe	jane.doe@customerA.cc	Global	a3b3f969-0272-43a9-8a

7. You will now verify that you can log in with the newly created user. First, you need to log out from the system as you are currently logged in as **admin**. Click the **User** icon in the top left corner and the logout icon will show up. Then click the **Logout** icon and when the Logout confirmation box shows up, confirm by clicking the **Logout** button.



8. At the login screen, enter the **observer1** user credentials, Tenant **CustomerA**, and click the **Log in** button.

Username: **observer1**

Password: **12345678**

Tenant: **CustomerA**

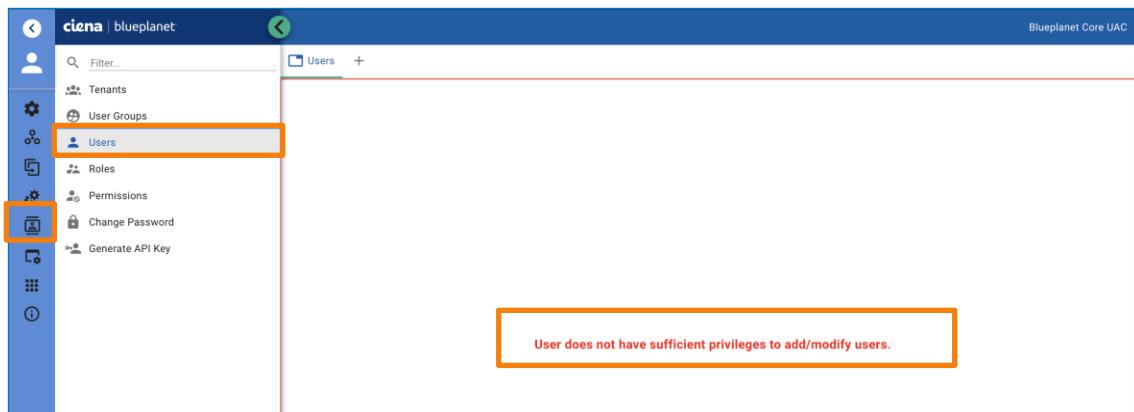
The form contains the following fields:

- * Username: observer1
- * Password: (redacted)
- Tenant: CustomerA

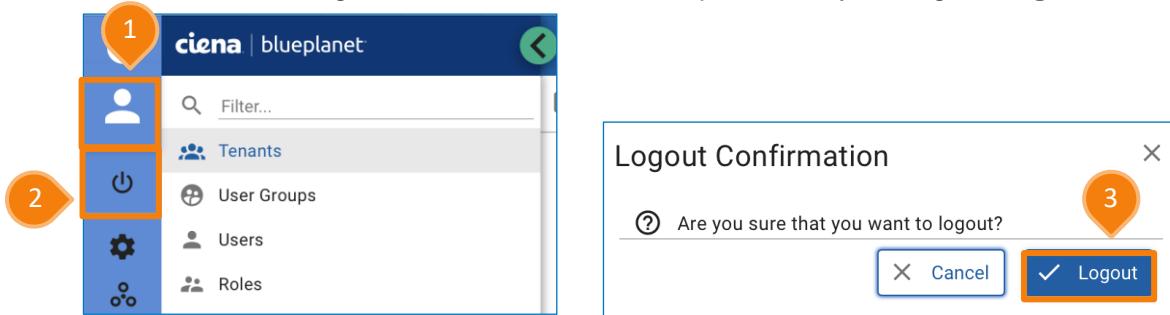
Buttons at the bottom:

- Forgot Password?
- Log in (highlighted with an orange border)

9. Click on the **UAC – Users** and verify that the observer1 user does not have sufficient privileges to add/modify users.



10. This was the last task for this lab, so you can log out of the system.
Click the **User** icon in the top left corner and the logout icon will show up. Then click the **Logout** icon and when the Logout Confirmation box shows up, confirm by clicking the **Logout** button.



End of Lab

Lab 2: Platform Operations and Maintenance

Objectives

- Get familiar with the Blue Planet K8s On-Prem installer and basic Helm commands
- Use Kubernetes commands to manage the cluster, nodes, services, and other objects
- Manage Platform backup and restore operations using snapshots
- Manage the health of your deployment by using Nagios
- Display built-in graphs and create custom graphs in Grafana
- Monitor Blue Planet solution logs using Kibana

Task 1: Blue Planet K8s On-Prem Installer

In this task, you will explore the Blue Planet K8s On-Prem installer script and learn some basic Helm commands to help you verify and perform operational tasks on your Kubernetes-based Blue Planet Platform.

1. From your Student VM desktop, open a **Terminal** window by clicking the **Terminal Emulator** icon in the panel located at the bottom of your desktop area.



2. Navigate to `/home/bpadmin/bpi` and list the contents of the directory.

```
[bpadmin@bp ~]$ cd /home/bpadmin/bpi
[bpadmin@bp bpi]$ ls -l
total 84
-rw-rw-r--. 1 bpadmin bpadmin   317 Apr  9 07:23 ansible.cfg
-rwxrwxr-x. 1 bpadmin bpadmin 22005 Apr  9 07:23 bpi
-rwxrwxr-x. 1 bpadmin bpadmin  4837 Apr  9 07:23 bp-onprem-install.sh
-rwxrwxr-x. 1 bpadmin bpadmin  3354 May 12 09:59 bp-onprem-install-
single-node.sh
-rwxrwxr-x. 1 bpadmin bpadmin 1819 Apr  9 07:23 bp-uninstall.sh
-rwxrwxr-x. 1 bpadmin bpadmin   121 Apr  9 07:23 cleanup.sh
-rwxrwxr-x. 1 bpadmin bpadmin   635 Apr  9 07:23 diff.sh
-rw-rw-r--. 1 bpadmin bpadmin  2096 May 12 09:53 hosts
-rw-rw-r--. 1 bpadmin bpadmin 2856 Apr  9 07:23 hosts.gr
-rw-rw-r--. 1 bpadmin bpadmin   534 Apr  9 07:23 LICENSE.md
drwxr-xr-x. 2 bpadmin bpadmin   131 May 24 12:14 logs
drwxrwxr-x. 6 bpadmin bpadmin  4096 Apr  9 07:23 playbooks
-rwxrwxr-x. 1 bpadmin bpadmin 1303 Apr  9 07:23 setup-bpi-standby.sh
```

```
-rwxrwxr-x. 1 badmin badmin 3944 Apr  9 07:23 setup.sh
-rwxrwxr-x. 1 badmin badmin 5295 Apr  9 07:23 setup-users.sh
drwxr-xr-x. 5 badmin badmin    77 May 12 09:52 venv
drwxrwxr-x. 6 badmin badmin   68 May 12 09:52 workspace
```

The Blue Planet Platform in your lab is installed on a single-node Kubernetes cluster, which is set up directly on your Student VM. The Platform installation was performed with the Blue Planet K8s On-Prem installer by running the **bpi** script with the **--k8s-setup-single-node** option.

3. List the contents and the line numbers of the Ansible hosts file in the bpi directory by running the **cat -n hosts** command. The relevant code is in lines **40-41**. You can see that the k8s-master-0 node is set to the localhost IP. Also notice that no additional hosts are defined under the **cluster** and **worker_nodes** sections. Your Student VM is the only target node for Ansible playbooks, executed by the Blue Planet Installer script.

```
[badmin@bp bpi]$ cat -n hosts
<...output omitted...
40 [master_node_primary]
41 k8s-master-0 ansible_host=127.0.0.1
<...output omitted...
```

4. List the contents and the line numbers of the bpi file by running the **cat -n bpi** command. The relevant code is in lines **274** and **380-381**. You can see that when bpi gets called with the **--k8s-setup-single-node** option, the Ansible playbook named **k8s-setup-single-node.yml** is executed.

```
[badmin@bp bpi]$ cat -n bpi
<...output omitted...
274     group.add_argument('--k8s-setup-single-node',
dest="k8ssetupsinglenode", help='setup kubernetes on a single node',
action='store_true')
<...output omitted...
380     elif args.k8ssetupsinglenode:
381         playbook = 'k8s-setup-single-node.yml'
<...output omitted...
```

5. The lineup deployed on your Blue Planet Platform is **lineup-soo**. This lineup was installed using the command **bpi --k8s-bp-onprem-deploy-single-node soo 22.2.3-77-single soo blueplanet**.

NOTE: Do not execute this command, the lineup is already installed on your Student VM.

Inspect the relevant code for deploying applications in the bpi script on lines 280-282 and 390-393. You can see that when bpi gets called with the **--k8s-bp-onprem-deploy-single-node** option, the **bp-onprem-install-single-node.sh** script is executed.

```
[badmin@bp bpi]$ cat -n bpi
<...output omitted...
280     group.add_argument('--k8s-bp-onprem-deploy-single-node',
nargs=4, metavar=('<k8s_lineup>', '<lineup_version>',
'<k8s_namespace>', '<helm_repository>'),
```

```

281                         dest="k8sbponpremdeployonenode",
282                         help='deploy bp apps on single node on-premise
   k8s')
<...output omitted...>
390     elif args.k8sbponpremdeployonenode:
391         subprocess.call(["%(rootdir)s/bp-onprem-install-single-
   node.sh" % globals(),
392                         args.k8sbponpremdeployonenode[0],
   args.k8sbponpremdeployonenode[1], args.k8sbponpremdeployonenode[2],
   args.k8sbponpremdeployonenode[3]])
393     sys.exit(0)

```

6. List the contents and the line numbers of the **bp-onprem-install-single.sh** script. You can see the four arguments that are passed to the script.

```
[bpadmin@bp bpi]$ cat -n bp-onprem-install-single-node.sh
1 #!/bin/bash
2
3  LINEUP=$1
4  VERSION=$2
5  NAMESPACE=$3
6  REPOSITORY=$4
```

7. The following commands add the bphub Helm repo, create the provided K8s namespace and create the docker-registry secret. To change the credentials for the Helm repo or the docker-registry, you could just delete them and add them with the new credentials.

```

8 # Add the blueplanet repo URL into the helm repo
9 helm repo add "$REPOSITORY"
https://bphub.blueplanet.com/artifactory/helm-local-dev/ --username
CUSTOMER1 --password
APIKEYAKCp8krL23gGrif8rPiEAEoEEinCLJBUp15Lr1rFDG
10
11 # Determine, create, and prepare the target k8s namespace
12 if [ "$NAMESPACE" != "default" ]; then
13     if ! kubectl get -o name "ns/$NAMESPACE" >/dev/null 2>&1; then
14         kubectl create ns "$NAMESPACE"
15     fi
16 fi
17 kubectl patch "ns/$NAMESPACE" -p "
18 {
```

```
19     \"metadata\": {  
20         \"labels\": {  
21             \"bp-namespace\": \"${NAMESPACE}\"  
22         }  
23     }  
24 }  
25 "  
26  
27 # Create Kubernetes Secret to authenticate the artifactory  
imagePullSecret  
28 kubectl -n "${NAMESPACE}" create secret docker-registry bp-site-  
registry-cred --docker-server=bphub.blueplanet.com --docker-  
username=CUSTOMER1 --docker-  
password=APIKEYAKCp8krL23gGrif8rPiEAEOEEinCLJBUp15Lr1rFDG
```

8. The next section installs the site-config chart from bphub and provides the inline variable values. This is where you would set the relevant attributes before installing Blue Planet applications.

```
30 # Install site-config helm chart from bphub  
31 helm -n "${NAMESPACE}" install site-config "$REPOSITORY/site-config"  
-f - <<-EOT  
32     podCIDR:      "10.244.0.0/16"  
33     serviceCIDR: "10.96.0.0/12"  
34  
35     imageRegistry:    bphub.blueplanet.com  
36     imagePullSecrets: bp-site-registry-cred  
37     imagePullPolicy:  Always  
38  
39     ha:        false  
40     multiAZ:   false  
41  
42     baseStorageDir: "/opt/ciena/${NAMESPACE}/bp2"  
43  
44     databaseUsers:  
45         super:  
46             username: "sbpadmin"  
47             password: "sbpadminpw"  
48         replication:  
49             username: "rbpadmin"
```

```
50      password: "rbpadminpw"
51      rewind:
52          username: "wbpadmin"
53          password: "wbpadminpw"
54      client:
55          username: "bpadmin"
56          password: "bpadminpw"
57
58  bpopgdatabaseUsers:
59      super:
60          username: "bpopgsuper"
61          password: "bpopgsuperpw"
62      replication:
63          username: "rbpadmin"
64          password: "rbpadminpw"
65      rewind:
66          username: "wbpadmin"
67          password: "wbpadminpw"
68      client:
69          username: "bpopguser"
70          password: "bpopguserpw"
71
72  smtpConfig:
73      username:           USERNAME
74      password:           PASSWORD
75      staticPath:         "/bp2/src/static"
76      requireAuthentication: false
77      requireTransportSecurity: false
78      host:                mdmail.ciena.com
79      defaultFromName:     Blue Planet Mailer
80      port:                25
81      defaultFromEmail:    bp2alerts@ciena.com
82 EOT
```

9. The following section installs the staticnbi chart. This is where you would provide the URL of the License Server. To change the URL of the License Server on an existing deployment, you could just reinstall the staticnbi chart with the new values.

```
84 # Install staticnbi helm chart from bphub
85 helm -n "$NAMESPACE" install licsvr-nbi "$REPOSITORY/staticnbi" -f
- <<-EOT
86     name: licsvr
87
88     attributes:
89         publish: true
90
91     providers:
92         - url: https://10.247.1.7:7071
93 EOT
```

10. Finally, these commands install the core-platform and the specific lineup, provided in the script arguments. The haproxy loadbalancer chart, which deploys the https LoadBalancer service, is also installed.

```
95 # Install core-platform helm chart from bphub
96 helm -n "$NAMESPACE" install core-platform "$REPOSITORY/core-
platform"
97
98 # Install respective lineup helm chart from bphub
99 if [[ "$VERSION" =~ "21.6" ]]; then
100     helm -n "$NAMESPACE" install "$LINEUP" "$REPOSITORY/lineup-
$LINEUP" --version "$VERSION" --
set=global.baseAppLogDir=/opt/ciena/bp2/log
101 elif [[ "$VERSION" =~ "22.2" ]]; then
102     helm -n "$NAMESPACE" install "$LINEUP" "$REPOSITORY/lineup-
$LINEUP" --version "$VERSION"
103 fi
104
105 # Install loadbalancer helm chart from bphub
106 helm -n "$NAMESPACE" install haproxy-lb "$REPOSITORY/haproxy-lb" --
set tcpPorts.bpcore-ssh.external=31622
```

11. At the end, the script executes a kubectl command to retrieve the external IP address of the https service.

```
108 # Get the https service
109 sleep 5
110 kubectl -n "$NAMESPACE" get svc/https
```

12. Run the **kubectl -n soo get svc/https** command to retrieve the external IP and TCP port of the https service. This is the information that you would use to connect to the Blue Planet Platform from your web browser or API client.

```
[bpadmin@bp bpi]$ kubectl -n soo get svc/https
NAME      TYPE      CLUSTER-IP      EXTERNAL-IP      PORT(S)
AGE
https    LoadBalancer  10.99.87.112  10.245.0.2
31622:30090/TCP,443:32088/TCP  21d
```

13. You already saw the helm install and helm repo add commands in the installer scripts. Run the **helm --help** command to view the available options of the helm command.
14. Run the **helm list -A** command to list all the installed charts.

```
[bpadmin@bp bpi]$ helm list -A
NAME        NAMESPACE      REVISION      UPDATED
STATUS      CHART          APP VERSION
core-platformsoo
+0100 BST deployed      1            2022-08-12 13:00:56.909190999
core-platform-22.8.0
haproxy-lb   soo           1            2022-08-12 13:01:04.935711137
+0100 BST deployed      haproxy-lb-22.8.0
licsvr-nbi   soo           1            2022-08-12 13:00:54.46206181
+0100 BST     deployed      staticnbi-22.8.0
site-config  soo           1            2022-08-12 13:00:51.852690486
+0100 BST     deployed      site-config-22.8.0
soo          soo           1            2022-08-12 13:01:00.473286945
+0100 BST     deployed      lineup-soo-22.2.3-82-single
```

15. List the installed Helm repositories by executing the **helm repo list** command.

```
[bpadmin@bp bpi]$ helm repo list
NAME      URL
blueplanet  https://bphub.blueplanet.com/artifactory/helm-local-dev/
```

16. Execute the **helm search repo --version 22.2** command to search the repo for all charts with the 22.2 version. You can also try the **-l** and **--devel** options.

NOTE: Your output for this and the following commands could be different, depending on the current contents of the bphub repository.

```
[bpadmin@bp bpi]$ helm search repo --version 22.2
NAME          CHART VERSION      APP VERSION
DESCRIPTION
blueplanet/app-KubernetesRA      22.2.1        4.0.2.2110
Blue Planet app for...
blueplanet/app-ac                 22.2.1        0.0.5
Blue Planet access controller (AC)
```

blueplanet/app-acpra	22.2.0	1.0.2.2202
Blue Planet app for acpra.		
blueplanet/app-api-crinoid	22.2.4	2.0.18
Blue Planet API Crinoid app		

17. Run the **helm search repo -l --devel -r lineup-soo** command to search for all SOO lineups.

NAME	CHART VERSION	APP VERSION	DESCRIPTION
blueplanet/lineup-soo	22.8.1-2		Blue Planet
soo multi host lineup			
blueplanet/lineup-soo	22.2.3-77-single		Blue
Planet soo single host lineup			
blueplanet/lineup-soo	22.2.3-76-single		Blue
Planet soo single host lineup			

18. You can download the chart from the repository to inspect it by running the **helm pull** command. Try downloading the chart which is already installed.

```
[bpadmin@bp bpi]$ helm pull blueplanet/lineup-soo --version 22.2.3-77-single
```

19. The **lineup-soo-22.2.3-77-single.tgz** archive is downloaded from bphub. You can try extracting and inspecting the contents of the archive.

```
[bpadmin@bp bpi]$ tar xzvf lineup-soo-22.2.3-77-single.tgz
```

20. A new **lineup-soo** directory is created. You can explore the contents of the directory, the charts, values, and template files.

```
[bpadmin@bp bpi]$ cd lineup-soo
[bpadmin@bp lineup-soo]$ ls -l
total 8
drwxrwxr-x 13 bpadmin bpadmin 287 Jun  6 14:48 charts
-rw-r--r--  1 bpadmin bpadmin 967 May  4 06:58 Chart.yaml
drwxrwxr-x  2 bpadmin bpadmin  23 Jun  6 14:48 templates
-rw-r--r--  1 bpadmin bpadmin  11 May  4 06:58 values.yaml
```

Task 2: Kubernetes Based Solution Operations

In this task, you will execute various Kubernetes commands to get familiar with the solution. You will also learn how to reset a specific pod in the event of software malfunctions.

1. From your Student VM desktop, open a **Terminal** window by clicking the **Terminal Emulator** icon in the panel located at the bottom of your desktop area.



2. In the terminal window, you will use the **kubectl** command with various options to manage your cluster. Typically, you will use a specific resource option with the kubectl command, and to get the list of available resources, use the **kubectl api-resources** command.

[bpadmin@bp ~]\$ kubectl api-resources			
NAME	SHORTNAMES	APIVERSION	
NAMESPACE	KIND		
bindings	v1		true
Binding			
componentstatuses	cs	v1	false
ComponentStatus			
configmaps	cm	v1	true
ConfigMap			
endpoints	ep	v1	true
Endpoints			
events	ev	v1	true
Event			
limitranges	limits	v1	true
LimitRange			
namespaces	ns	v1	false
Namespace			
nodes	no	v1	false
Node			
persistentvolumeclaims	pvc	v1	true
PersistentVolumeClaim			
persistentvolumes	pv	v1	false
PersistentVolume			
pods	po	v1	true
Pod			
podtemplates		v1	true
PodTemplate			
replicationcontrollers	rc	v1	true
ReplicationController			
resourcequotas	quota	v1	true
ResourceQuota			
secrets		v1	true
Secret			
serviceaccounts	sa	v1	true
ServiceAccount			

services	svc	v1	true
Service			
mutatingwebhookconfigurations		admissionregistration.k8s.io/v1	false
MutatingWebhookConfiguration			
validatingwebhookconfigurations		admissionregistration.k8s.io/v1	false
ValidatingWebhookConfiguration			
customresourcedefinitions	crd, crds	apiextensions.k8s.io/v1	false
CustomResourceDefinition			
apiservices		apiregistration.k8s.io/v1	false
APIService			
controllerrevisions		apps/v1	true
ControllerRevision			
daemonsets	ds	apps/v1	true
DaemonSet			
deployments	deploy	apps/v1	true
Deployment			
replicasets	rs	apps/v1	true
ReplicaSet			
statefulsets	sts	apps/v1	true
StatefulSet			
tokenreviews		authentication.k8s.io/v1	false
TokenReview			
localsubjectaccessreviews		authorization.k8s.io/v1	true
LocalSubjectAccessReview			
selfsubjectaccessreviews		authorization.k8s.io/v1	false
SelfSubjectAccessReview			
selfsubjectrulesreviews		authorization.k8s.io/v1	false
SelfSubjectRulesReview			
subjectaccessreviews		authorization.k8s.io/v1	false
SubjectAccessReview			
horizontalpodautoscalers	hpa	autoscaling/v1	true
HorizontalPodAutoscaler			
cronjobs	cj	batch/v1	true
CronJob			
jobs		batch/v1	true
Job			
certificatesigningrequests	csr	certificates.k8s.io/v1	false
CertificateSigningRequest			
leases		coordination.k8s.io/v1	true
Lease			
endpointslices		discovery.k8s.io/v1	true
EndpointSlice			
events	ev	events.k8s.io/v1	true
Event			
ingresses	ing	extensions/v1beta1	true
Ingress			
flowschemas		flowcontrol.apiserver.k8s.io/v1beta1	false
FlowSchema			
prioritylevelconfigurations		flowcontrol.apiserver.k8s.io/v1beta1	false
PriorityLevelConfiguration			

rawdevices		nativestor.alauda.io/v1	false
RawDevice			
ingressclasses		networking.k8s.io/v1	false
IngressClass			
ingresses	ing	networking.k8s.io/v1	true
Ingress			
networkpolicies	netpol	networking.k8s.io/v1	true
NetworkPolicy			
runtimeclasses		node.k8s.io/v1	false
RuntimeClass			
poddisruptionbudgets	pdb	policy/v1	true
PodDisruptionBudget			
podsecuritypolicies	psp	policy/v1beta1	false
PodSecurityPolicy			
clusterrolebindings		rbac.authorization.k8s.io/v1	false
ClusterRoleBinding			
clusterroles		rbac.authorization.k8s.io/v1	false
ClusterRole			
rolebindings		rbac.authorization.k8s.io/v1	true
RoleBinding			
roles		rbac.authorization.k8s.io/v1	true
Role			
priorityclasses	pc	scheduling.k8s.io/v1	false
PriorityClass			
csidrivers		storage.k8s.io/v1	false
CSIDriver			
csinodes		storage.k8s.io/v1	false
CSINode			
csistoragecapacities		storage.k8s.io/v1beta1	true
CSIStorageCapacity			
storageclasses	sc	storage.k8s.io/v1	false
StorageClass			
volumeattachments		storage.k8s.io/v1	false
VolumeAttachment			
logicalvolumes		topolvm.cybozu.com/v1	false
LogicalVolume			
topolvmclusters		topolvm.cybozu.com/v2	true
TopolvmCluster			
[bpadmin@bp ~]\$			

3. To display the documentation on the specific resource and its fields, use the **kubectl explain** command. For example, to list the documentation for the resource containers, use the command with the **pods.spec.containers** option.

```
[bpadmin@bp ~]$ kubectl explain pods.spec.containers
I0606 13:55:04.135069 27477 request.go:668] Waited for 1.182786716s
due to client-side throttling, not priority and fairness, request:
GET:https://127.0.0.1:6443/apis/policy/v1?timeout=32s
KIND: Pod
VERSION: v1
```

RESOURCE: containers <[]Object>

DESCRIPTION:

List of containers belonging to the pod. Containers cannot currently be added or removed. There must be at least one container in a Pod. Cannot be updated.

A single application container that you want to run within a pod.

FIELDS:

args <[]string>

Arguments to the entrypoint. The docker image's CMD is used if this is not

provided. Variable references `$(VAR_NAME)` are expanded using the container's environment. If a variable cannot be resolved, the reference in

the input string will be unchanged. The `$(VAR_NAME)` syntax can be escaped

with a double `$$`, ie: `$$$(VAR_NAME)`. Escaped references will never be

expanded, regardless of whether the variable exists or not. Cannot be

updated. More info:

<https://kubernetes.io/docs/tasks/inject-data-application/define-command-argument-container/#running-a-command-in-a-shell>

command <[]string>

Entrypoint array. Not executed within a shell. The docker image's ENTRYPOINT is used if this is not provided. Variable references `$(VAR_NAME)`

are expanded using the container's environment. If a variable cannot be

resolved, the reference in the input string will be unchanged. The `$(VAR_NAME)` syntax can be escaped with a double `$$`, ie: `$$$(VAR_NAME)`.

Escaped references will never be expanded, regardless of whether the

variable exists or not. Cannot be updated. More info:

<https://kubernetes.io/docs/tasks/inject-data-application/define-command-argument-container/#running-a-command-in-a-shell>

env <[]Object>

List of environment variables to set in the container. Cannot be updated.

<...output omitted...>

4. Verify that the user you are currently logged in as has full privileges to run kubectl commands. To do that, use the **kubectl auth can-i** command with the options '*' and '*'.

```
[bpadmin@bp ~]$ kubectl auth can-i '*' '*'  
yes
```

5. Now examine the Blue Planet Platform solution deployed on your pod. First, use the **kubectl describe node** command to get the information on the node. Take a minute to examine the detailed output.

```
[bpadmin@bp ~]$ kubectl describe node  
Name:           bp  
Roles:          control-plane, master  
Labels:         beta.kubernetes.io/arch=amd64  
                beta.kubernetes.io/os=linux  
                kubernetes.io/arch=amd64  
                kubernetes.io/hostname=bp  
                kubernetes.io/os=linux  
                node-role.kubernetes.io/control-plane=  
                node-role.kubernetes.io/master=  
                node.kubernetes.io/exclude-from-external-load-  
                balancers=  
                topology.topolvm.cybozu.com/node=bp  
Annotations:    capacity.topolvm.cybozu.com/00default: 1546184032256  
                capacity.topolvm.cybozu.com/ssd: 1546184032256  
                csi.volume.kubernetes.io/nodeid:  
                {"topolvm.cybozu.com":"bp"}  
                flannel.alpha.coreos.com/backend-data:  
                {"VNI":1,"VtepMAC":"92:17:28:cb:0b:1c"}  
                flannel.alpha.coreos.com/backend-type: vxlan
```

```
        flannel.alpha.coreos.com/kube-subnet-manager: true
        flannel.alpha.coreos.com/public-ip: 10.156.30.51
        kubeadm.alpha.kubernetes.io/cri-socket:
        /var/run/dockershim.sock
            node.alpha.kubernetes.io/ttl: 0
            volumes.kubernetes.io/controller-managed-attach-
            detach: true
        CreationTimestamp: Mon, 16 May 2022 12:38:23 +0100
        Taints: <none>
        Unschedulable: false

    ... (output omitted)

    Addresses:
        InternalIP: 10.154.12.11
        Hostname: bp

    Capacity:
        cpu: 12
        ephemeral-storage: 218502Mi
        hugepages-1Gi: 0
        hugepages-2Mi: 0
        memory: 30714924Ki
        pods: 243

    Allocatable:
        cpu: 11250m
        ephemeral-storage: 204056873848
        hugepages-1Gi: 0
        hugepages-2Mi: 0
        memory: 28105772Ki
        pods: 243

    System Info:
        Machine ID: 337110c292db47689f97adc4bc1121ea
        System UUID: 2BF32342-1EDD-9A4D-AD06-C7B6B217E3E5
        Boot ID: e87cf819-a3f9-441d-9d94-1a2e4fe8d5ee
        Kernel Version: 3.10.0-1160.62.1.el7.x86_64
        OS Image: CentOS Linux 7 (Core)
```

Operating System:	linux
Architecture:	amd64
Container Runtime Version:	docker://19.3.15
Kubelet Version:	v1.21.8
Kube-Proxy Version:	v1.21.8
PodCIDR:	10.244.0.0/24
PodCIDRs:	10.244.0.0/24

6. Next, use the **kubectl cluster-info** command to display the cluster information.

```
[bpadmin@bp ~]$ kubectl cluster-info
Kubernetes control plane is running at https://127.0.0.1:6443
CoreDNS is running at https://127.0.0.1:6443/api/v1/namespaces/kube-
system/services/kube-dns:dns/proxy
```

To further debug and diagnose cluster problems, use 'kubectl cluster-info dump'.

NOTE: To get a very detailed output, run the **kubectl cluster-info dump** command.

7. From the terminal window, list the Kubernetes namespaces using the **kubectl get namespaces** command.

```
[bpadmin@bp ~]$ kubectl get namespaces
NAME        STATUS   AGE
default     Active   8d
kube-node-lease  Active   8d
kube-public    Active   8d
kube-system    Active   8d
metallb-system Active   8d
nativestor-system Active   8d
soo          Active   8d
[bpadmin@bp ~]$
```

8. List the deployments in the soo namespace using the **kubectl get deployments -n soo** command.

```
[bpadmin@bp ~]$ kubectl get deployments -n soo
NAME      READY   UP-TO-DATE   AVAILABLE   AGE
bp-ac    1/1     1           1           21d
```

9. Now list the nodes in your Kubernetes deployment with the **kubectl get nodes** command.

```
[bpadmin@bp ~]$ kubectl get nodes
NAME      STATUS   ROLES   AGE   VERSION
[redacted]
```

bp	Ready	control-plane,master	21d	v1.21.8
----	-------	----------------------	-----	---------

NOTE: In your lab, there is only one server in the cluster. In production environments, you might see multiple worker nodes, depending on the deployment.

10. Next, from the terminal window, list all the pods in the deployed solution using the **kubectl get pods** command. Use the **-A** option to list all the pods from all namespaces. Note that the pods belong to different namespaces from the list of namespaces as you listed them in the previous step.

[bpadmin@bp ~]\$ kubectl get pods -A			
NAMESPACE	NAME	READY	STATUS
RESTARTS	AGE		
kube-system	coredns-558bd4d5db-9kqs2	1/1	
Running	3 8d		
kube-system	coredns-558bd4d5db-c2k57	1/1	
Running	3 8d		
kube-system	etcd-bp	1/1	Running
3 8d			
kube-system	kube-apiserver-bp	1/1	Running
4 8d			
kube-system	kube-controller-manager-bp	1/1	
Running	8 8d		
kube-system	kube-flannel-ds-phzcf	1/1	
Running	10 8d		
kube-system	kube-proxy-l8t2n	1/1	Running
3 8d			
kube-system	kube-scheduler-bp	1/1	Running
8 8d			
metallb-system	controller-7dcc8764f4-vk4dj	1/1	
Running	1 25h		
metallb-system	speaker-w9j7k	1/1	Running
1 25h			
nativestor-system	discover-device-wnpbf	1/1	
Running	3 8d		
nativestor-system	nativestor-6f79d9ff58-sv2cw	1/1	
Running	3 8d		
nativestor-system	topolvm-controller-687fbfdbf6-hbgc2	4/4	
Running	20 8d		
nativestor-system	topolvm-controller-687fbfdbf6-r9811	4/4	
Running	19 8d		
nativestor-system	topolvm-node-bp-5c87486b5c-qkvbt	4/4	
Running	14 8d		
nativestor-system	topolvm-prepare-vg-226a193ae269296d44f3efd3d66fead5-6zfvk	0/1	
Completed	0 19h		
soo	api-crinoid-0	1/1	Running
3 8d			
soo	api-gw-0	1/1	Running
3 8d			
soo	backup-service-0	1/1	Running
6 8d			

soo		bp-ac-75c558bc49-csphw	1/1	Running
soo	3	8d bp-platform-ui-0	1/1	Running
soo	3	8d bpaudit-0	1/1	Running
soo	3	8d bplice-0	1/1	Running
soo	3	8d bpmn-0	1/1	Running
soo	3	8d bpo-admin-guide-0	1/1	Running
soo	3	8d bpo-ui-0	1/1	Running
soo	3	8d bpo-user-guide-0	1/1	Running
soo	3	8d bpocore-0	1/1	Running
soo	3	8d bpogg-0	1/1	Running
...(output omitted)				

NOTE: To get the list of pods along with automatically generated labels, use the **kubectl get pods -A --show-labels** command.

- To list the replicaset, use the **kubectl describe rs** command.

```
[bpadmin@bp ~]$ kubectl describe rs -n soo
Name:           bp-ac-75c558bc49
Namespace:      soo
Selector:       bp-ac=adm,pod-template-hash=75c558bc49
Labels:         bp-ac=adm
                pod-template-hash=75c558bc49
Annotations:    deployment.kubernetes.io/desired-replicas: 1
                deployment.kubernetes.io/max-replicas: 2
                deployment.kubernetes.io/revision: 1
                meta.helm.sh/release-name: core-platform
                meta.helm.sh/release-namespace: soo
Controlled By: Deployment/bp-ac
Replicas:       1 current / 1 desired
Pods Status:    1 Running / 0 Waiting / 0 Succeeded / 0 Failed
Pod Template:
  Labels:  bp-ac=adm
          pod-template-hash=75c558bc49
Containers:
```

```

app:
  Image:      bphub.blueplanet.com/blueplanet/ac:0.0.5
  Port:       4443/TCP
  Host Port: 0/TCP
  Command:
    ac
  Environment: <none>
  Mounts:
    /dev/log from syslog (rw)
    /etc/bp2/site-config from site-config (ro)
    /root/cert from cert (ro)
  Volumes:
    syslog:
      Type:        HostPath (bare host directory volume)
      Path:        /dev/log
      HostPathType: Socket
    site-config:
      Type:        ConfigMap (a volume populated by a ConfigMap)
      Name:        site-config
      Optional:   false
    cert:
      Type:        Secret (a volume populated by a Secret)
      SecretName: bp-ac
      Optional:   false
  Events:     <none>
[bpadmin@bp ~]$
```

12. If you would like to limit the scope of the pods to a specific namespace, use the **-n** option with the name of the namespace. List the pods belonging to the **metallb-system** namespace:

```
[bpadmin@bp ~]$ kubectl get pods -n metallb-system
NAME                  READY   STATUS    RESTARTS   AGE
controller-7dcc8764f4-vk4dj  1/1     Running   1          26h
speaker-w9j7k           1/1     Running   1          26h
[bpadmin@bp ~]$
```

13. To list all services and respective cluster IPs, external IPs, and ports, use the **kubectl get svc -A** command.

```
[bpadmin@bp ~]$ kubectl get svc -A
```

NAMESPACE	NAME	TYPE	CLUSTER-IP
EXTERNAL-IP	PORT(S)	AGE	
default	kubernetes	ClusterIP	10.96.0.1
<none>	443/TCP	8d	
kube-system	kube-dns	ClusterIP	10.96.0.10
<none>	53/UDP,53/TCP,9153/TCP	8d	
nativestor-system	nativestor-cluster-metric	ClusterIP	
10.103.168.41	<none>	8080/TCP	8d
soo	api-controller	ClusterIP	10.99.34.4
<none>	8000/TCP	8d	
soo	apigwinspect	ClusterIP	
10.101.186.230	<none>	8090/TCP	8d
soo	asset-manager	ClusterIP	
10.104.207.59	<none>	8181/TCP	8d
soo	assets	ClusterIP	
10.109.214.74	<none>	80/TCP	8d
soo	auditreport	ClusterIP	10.99.226.74
<none>	5000/TCP	8d	
soo	backup-aep	ClusterIP	
10.103.94.155	<none>	8080/TCP	8d
soo	backupservice	ClusterIP	
10.99.112.162	<none>	8080/TCP	8d
soo	blueplanet-app-bar-ui	ClusterIP	
10.107.203.71	<none>	8080/TCP	8d
soo	blueplanet-bpmn-rest-showcase	ClusterIP	
10.99.177.30	<none>	8080/TCP	8d
soo	blueplanet-bpmn-ui	ClusterIP	
10.110.100.177	<none>	8080/TCP	8d
soo	blueplanet-core-common	ClusterIP	10.103.2.85
<none>	8080/TCP	8d	
soo	blueplanet-core-prefs	ClusterIP	10.99.2.177
<none>	8080/TCP	8d	
soo	blueplanet-core-showcase	ClusterIP	
10.111.95.140	<none>	8080/TCP	8d
soo	blueplanet-core-showcase-rest-internal	ClusterIP	
10.106.142.248	<none>	8080/TCP	8d
soo	blueplanet-core-uac	ClusterIP	
10.102.75.108	<none>	8080/TCP	8d
soo	bp-ac	ClusterIP	10.108.32.143
<none>	443/TCP	8d	
<...output omitted...>			
soo	https	LoadBalancer	10.99.87.112
10.245.0.2	31622:30090/TCP,443:32088/TCP	8d	
<...output omitted...>			

14. Note that one service (https) is of a type **LoadBalancer**. This service has an external IP address assigned to it and this IP is used to access the Blue Planet UI over HTTPS. To learn the external IP of your Blue Planet deployment, use the command:

```
[bpadmin@bp ~]$ kubectl describe svc https -n soo
Name:           https
```

```

Namespace:          soo
Labels:             app.kubernetes.io/managed-by=Helm
Annotations:        meta.helm.sh/release-name: haproxy-lb
                     meta.helm.sh/release-namespace: soo
Selector:           bp-app=haproxy
Type:               LoadBalancer
IP Family Policy:  SingleStack
IP Families:       IPv4
IP:                10.99.87.112
IPs:               10.99.87.112
LoadBalancer Ingress:  10.245.0.2
Port:               bpocore-ssh 31622/TCP
TargetPort:         31622/TCP
NodePort:           bpocore-ssh 30090/TCP
Endpoints:         10.244.0.215:31622
Port:               https 443/TCP
TargetPort:         443/TCP
NodePort:           https 32088/TCP
Endpoints:         10.244.0.215:443
Session Affinity:  None
External Traffic Policy: Cluster
Events:             <none>

```

15. To find out the IP range that the load balancer uses in your pod, use the following command:

```

[bpadmin@bp ~]$ kubectl describe cm -n metallb-system config
Name:         config
Namespace:    metallb-system
Labels:       <none>
Annotations: <none>

Data
====

config:
-----
address-pools:
- name: default

```

```
protocol: layer2
addresses:
- "10.245.0.2-10.245.0.5"
```

Events: <none>

16. Occasionally, in the life of your application, a pod can get stuck or become unresponsive. In that case, you can reset the affected pod from the CLI. To reset a specific pod, use the **kubectl delete** command. The Kubernetes solution monitors pods and as soon as a pod is deleted, the system will bring another pod up with an identical initial configuration for that pod. Delete the **fcweb-0** pod which is the https server for the platform. Your UI will experience an outage until the pod gets redeployed by the system.

```
[bpadmin@bp ~]$ kubectl delete pod -n soo fcweb-0 --force
warning: Immediate deletion does not wait for confirmation that the
running resource has been terminated. The resource may continue to
run on the cluster indefinitely.

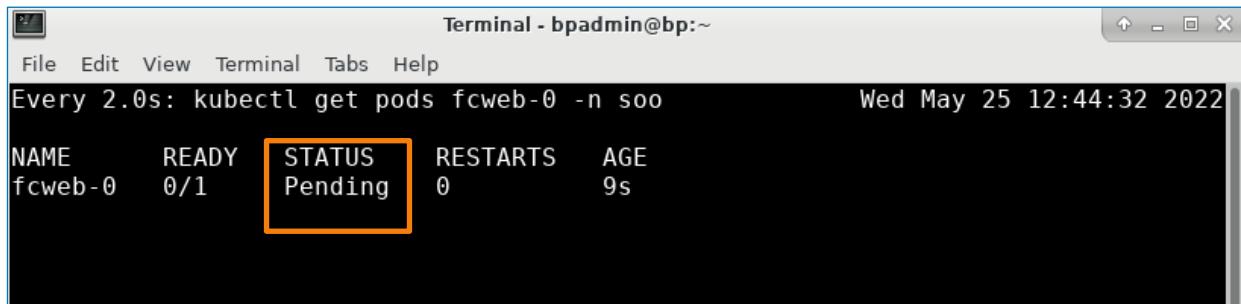
pod "fcweb-0" force deleted
```

NOTE: Your Blue Planet UI is now unresponsive while the pod is restarting.

17. Now, execute the **watch 'kubectl get pods fcweb-0 -n soo'** command to monitor the state of the pod.

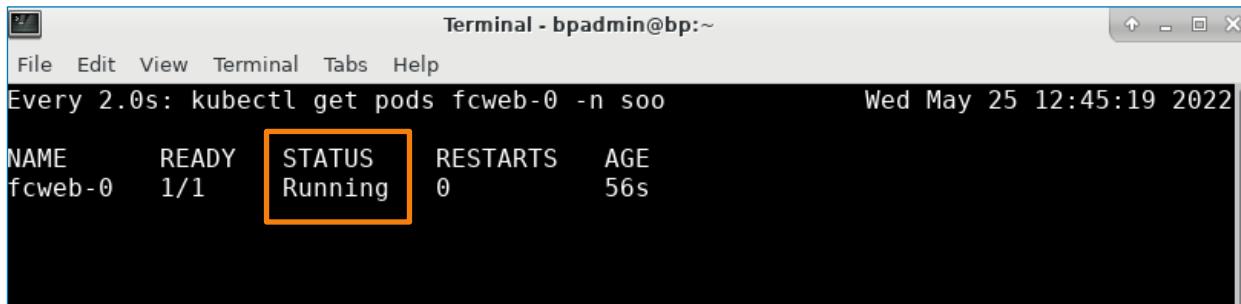
```
[bpadmin@bp ~]$ watch 'kubectl get pods fcweb-0 -n soo'
```

18. The watch command will execute the **kubectl get pods** command every 2 seconds. Monitor the state of the pod in your terminal window. When the state transitions from **Pending** to **Running**, you can exit the watch command by pressing the **CTRL+C** key combination.



Terminal - bpadmin@bp:~

```
File Edit View Terminal Tabs Help
Every 2.0s: kubectl get pods fcweb-0 -n soo              Wed May 25 12:44:32 2022
NAME      READY   STATUS    RESTARTS   AGE
fcweb-0   0/1     Pending   0          9s
```



Terminal - bpadmin@bp:~

```
File Edit View Terminal Tabs Help
Every 2.0s: kubectl get pods fcweb-0 -n soo              Wed May 25 12:45:19 2022
NAME      READY   STATUS    RESTARTS   AGE
fcweb-0   1/1     Running   0          56s
```

NOTE: It might take a few extra minutes for your Blue Planet UI to become responsive again.

Task 3: Backup and Restore

In this task, you will use Web UI and Node Solution Manager to perform the backup and restore operations. You will create a schedule to perform periodic backups. You will trigger a backup snapshot creation. You will download a snapshot to the Student VM host and provide an existing snapshot to the backup and restore solution.

1. Log in again to the Blue Planet Platform UI, using the Chrome browser from within your Student PC session (note the shortcut on the remote desktop screen) and enter <https://bp.lab.local> in the address space, and press **Enter**.

At the login screen, enter the *admin* user credentials and Tenant *master*, and click the **Login** button.

Username: **admin**

Password: **adminpw**

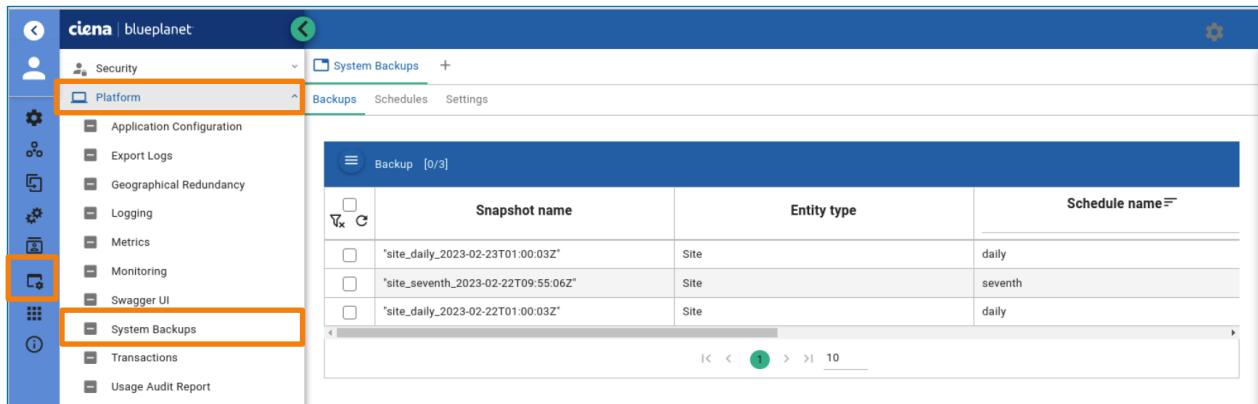
Tenant: **master**

The screenshot shows the Blue Planet Platform login interface. It features a logo with the letters 'bp' in a blue circle at the top. Below the logo are three input fields: 'Username' (containing 'admin'), 'Password' (containing '.....'), and 'Tenant' (containing 'master'). At the bottom of the form are two buttons: 'Forgot Password?' and a blue 'Log in' button.

2. In the next step, you will be asked to enter the Subtenant. Leave this field blank and just click on the **Apply** button.

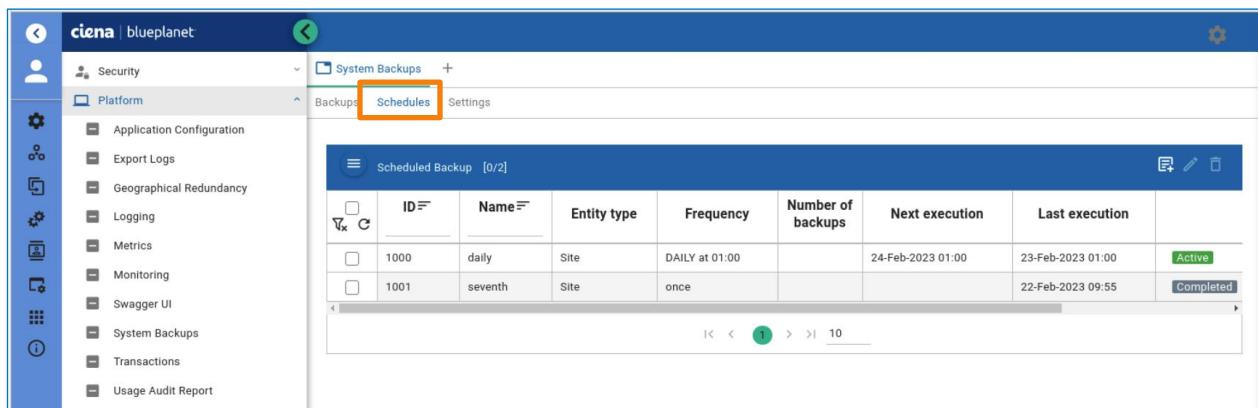
The screenshot shows a dialog box for selecting a subtenant. It features a logo with 'bp' in a blue circle at the top. Below the logo is a dropdown menu labeled 'Subtenant:' with a small downward arrow icon. At the bottom of the dialog is a blue 'Apply' button.

3. From your browser Blue Planet UI session, select **System > Platform > System Backups** to access the Backup solution in the UI.



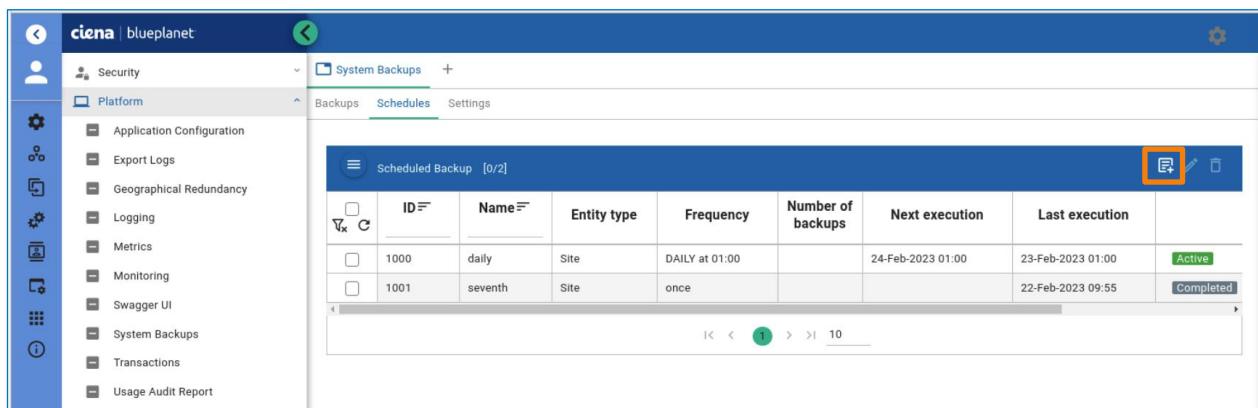
Snapshot name	Entity type	Schedule name
'site_daily_2023-02-23T01:00:03Z'	Site	daily
'site_seventh_2023-02-22T09:55:06Z'	Site	seventh
'site_daily_2023-02-22T01:00:03Z'	Site	daily

4. Open the **Schedules** tab. You will see a list of existing schedules, their parameters, and their state.



ID	Name	Entity type	Frequency	Number of backups	Next execution	Last execution	
1000	daily	Site	DAILY at 01:00		24-Feb-2023 01:00	23-Feb-2023 01:00	Active
1001	seventh	Site	once			22-Feb-2023 09:55	Completed

5. Create a new Backup Schedule by clicking the **Backup** button in the upper right corner of the **Schedule** tab.



ID	Name	Entity type	Frequency	Number of backups	Next execution	Last execution	
1000	daily	Site	DAILY at 01:00		24-Feb-2023 01:00	23-Feb-2023 01:00	Active
1001	seventh	Site	once			22-Feb-2023 09:55	Completed

6. Use the following parameters in the new schedule:

Schedule name: **Weekly**

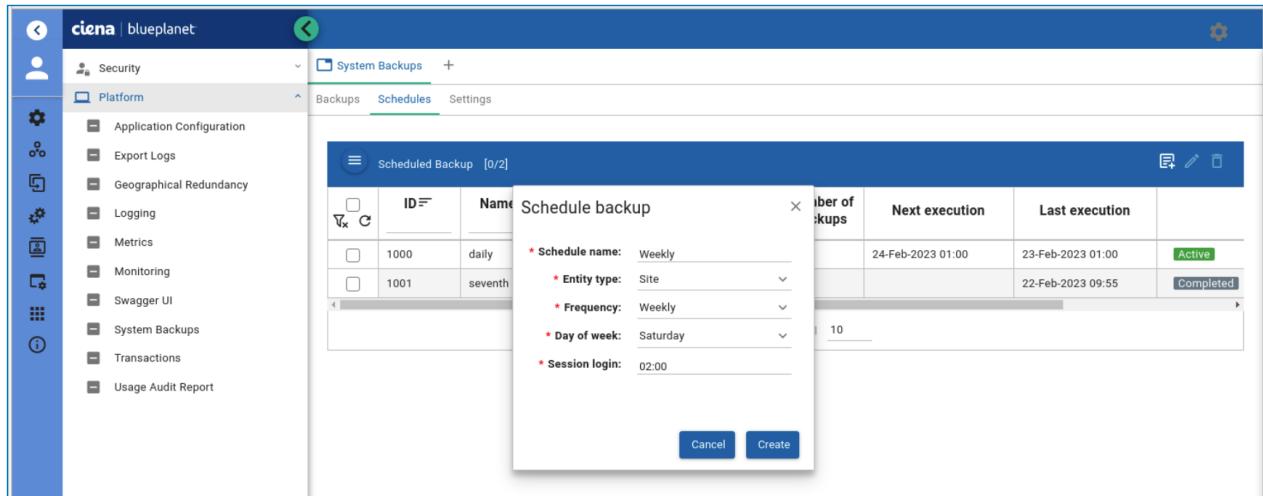
Entity type: **Site**

Frequency: **Weekly**

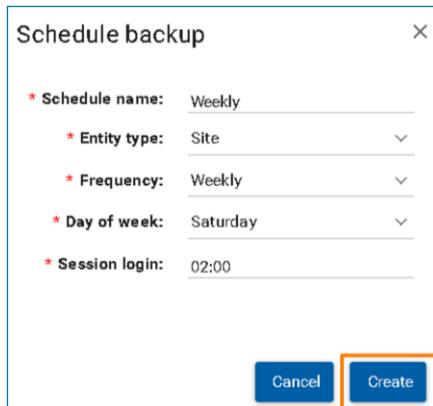
Day of week: **Saturday**

Session login: **02:00**

This schedule will automatically create a backup snapshot every Saturday at 02:00 hours.



- Press **Create** when finished to confirm the settings of the new schedule.



* Schedule name:	Weekly
* Entity type:	Site
* Frequency:	Weekly
* Day of week:	Saturday
* Session login:	02:00

- Create another Backup schedule by clicking the **Backup** button.
- This time select the following parameters of the new schedule:

Schedule name: **Snapshot1**

Entity type: **Site**

Frequency: **Now**

This schedule will create a backup snapshot once, and it will do it immediately after the creation of the schedule.

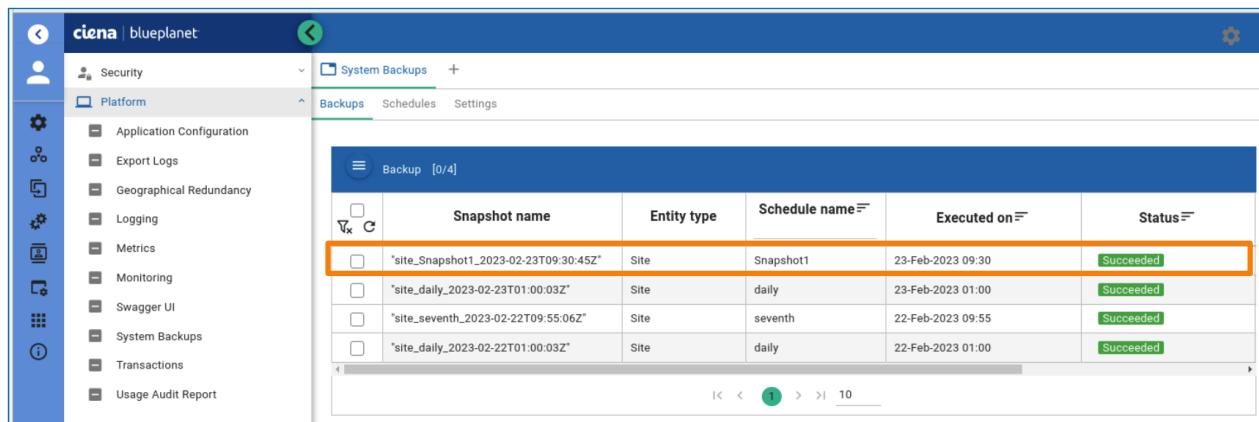
Schedule backup

* Schedule name: Snapshot1

* Entity type: Site

* Frequency: Now

10. Go to the **Backups** tab and look at the list of existing backup snapshots. You will see a new snapshot created by the new schedule. If the Status of the snapshot shows “In Progress”, that means that the snapshot is still being created. In this case, wait a few moments and refresh the browser window. When the snapshot creation is done, the status should show “Succeeded”.



	Snapshot name	Entity type	Schedule name	Executed on	Status
<input type="checkbox"/>	"site_Snapshot1_2023-02-23T09:30:45Z"	Site	Snapshot1	23-Feb-2023 09:30	Succeeded
<input type="checkbox"/>	"site_daily_2023-02-23T01:00:03Z"	Site	daily	23-Feb-2023 01:00	Succeeded
<input type="checkbox"/>	"site_seventh_2023-02-22T09:55:06Z"	Site	seventh	22-Feb-2023 09:55	Succeeded
<input type="checkbox"/>	"site_daily_2023-02-22T01:00:03Z"	Site	daily	22-Feb-2023 01:00	Succeeded

11. Open a **Terminal** window on the Student VM.
12. Find the NSM container by using the **kubectl get pods** command. Search through all namespaces (parameter **-A**), use the wide output to also see the IP addresses (option **-o wide**), and use the filter to search for the NSM container (use pipe with grep nsm). The important information from the output is the container ID, the namespace that includes the NSM container, and the IP address of the container.

```
$ kubectl get pods -A -o wide | grep nsm
soo      nsm-r25rd   1/1     Running    4       21d    10.244.0.175   bp
```

13. Connect to the NSM container by using the **namespace** and **container ID** parameters gathered in the previous step.

```
$ kubectl exec -it -n soo nsm-r25rd -- bash
root@nsm-r25rd:~#
```

14. List all the snapshots that are already available in the NSM container by creating an API call using a curl command. Use the “jq” tool to format the output to be easily readable. You will notice that the last snapshot is the one made in steps 6 and 7.

```
root@nsm-r25rd:~# curl localhost/api/v1/snapshots | jq
{
  "kind": "#snapshots",
```

```
"nextPageToken": null,  
"itemsAvailable": 4,  
"items": [  
  {  
    "path": "/bp2/snapshots/site_daily_2023-02-22T01:00:03Z.snap",  
    "name": "site_daily_2023-02-22T01:00:03Z",  
    "solution": "site",  
    "solution_type": "site",  
    "label": "daily",  
    "timestamp": "2023-02-22T01:00:03Z",  
    "files": [      ".",  
      "./api-crinoid-0",  
      "./backup-service-0",  
      "./bpocore-0",  
      "./bpopg-0",  
      "./fccustartifacts-0",  
      "./gcs-0",  
      "./kafka-0",  
      "./open-api-0",  
      "./postgres-0"  
    ]  
  }]
```

15. Create a new snapshot using an API call with **curl -X put localhost/api/v1/site/backup**. A backup snapshot will be created, and the prompt output will show the name of the snapshot.

```
root@nsm-r25rd:~# curl -X put localhost/api/v1/site/backup  
"site_unlabeled_2023-02-23T09:44:56Z"
```

16. Exit the NSM container. Download the snapshot from the NSM container to the Student VM host. by using the following command:

```
curl -o <local filename> http://<NSM IP>/api/v1/snapshots/<snapshot name>/download
```

Use the same local filename as the filename of the snapshot.

Use the IP address of the NSM container found in step 10.

You will have to use root permissions to execute this command.

```
root@nsm-r25rd:~# exit  
$ sudo curl -o site_unlabeled_2023-02-23T09:44:56Z.snap  
http://10.244.0.175/api/v1/snapshots/site_unlabeled_2023-02-  
23T09:44:56Z/download
```

% Total	% Received	% Xferd	Average Speed	Time	Time	Time	
Current			Dload	Upload	Total	Spent	Left
Speed							
100	12.1M	0	12.1M	0	0	4003k	0
					--:--:--	0:00:03	--:--:--
4004k							

17. Print the output of the `/home/bpadmin` directory. Look for a snapshot file `site_unlabeled_2023-02-23T09:44:56Z.snap`

```
[bpadmin@bp ~]$ ll
total 73004
drwxrwxr-x. 7 bpadmin bpadmin 4096 May 12 2022 bpi
-rw-rw-r--. 1 bpadmin bpadmin 49828117 Apr 11 2022 bpi-k8s-22.02-3.sh
-rwxrwxr-x 1 bpadmin bpadmin 369 May 16 2022 config-k8s-
metallb.sh
drwxrwxr-x. 2 bpadmin bpadmin 29 Jun 7 2022 Desktop
drwx----- 2 bpadmin bpadmin 6 May 16 2022 Downloads
drwxr-xr-x 2 bpadmin bpadmin 82 Aug 12 2022 nativestor-pv
drwxrwxr-x 3 bpadmin bpadmin 56 May 16 2022 openldap
drwxrwxr-x 3 bpadmin bpadmin 19 May 16 2022 Postman
-rw-r--r-- 1 root root 12431360 Feb 23 09:55 site_daily_2023-02-
21T01:00:03Z.snap
-rw-r--r-- 1 root root 12482560 Feb 23 09:50 site_unlabeled_2023-
02-23T09:44:56Z.snap
```

18. Upload the snapshot `site_daily_2023-02-21T01:00:03Z.snap` to the NSM container by using the following API call:

```
curl -i -X POST --data-binary "@<local filename>" http://<NSM
IP>/api/v1/snapshots/upload?snapshot=<snapshot name>
```

```
$ curl -i -X post --data-binary "@site_daily_2023-02-21T01:00:03Z.snap"
http://10.244.0.175/api/v1/snapshots/upload?snapshot=site_daily_2023-
02-21T01:00:03Z
HTTP/1.1 100 Continue

HTTP/1.1 200 OK
Server: TwistedWeb/22.4.0
Date: Thu, 23 Feb 2023 10:02:29 GMT
Content-Length: 370
Content-Type: text/html
```

```
{"path": "/bp2/snapshots/site_daily_2023-02-21T01:00:03Z.snap", "name": "site_daily_2023-02-21T01:00:03Z", "solution": "site", "solution_type": "site", "label": "daily", "timestamp": "2023-02-21T01:00:03Z", "files": [".", "./api-crinoid-0", "./backup-service-0", "./bpocore-0", "./bpopg-0", "./fccustartifacts-0", "./gcs-0", "./kafka-0", "./open-api-0", "./postgres-0"]}
```

19. Connect to the NSM container again and list the snapshots and their content. You should see the snapshot that you just uploaded at the end of the output.

```
$ kubectl exec -it -n sooo nsm-r25rd -- bash  
root@nsm-r25rd:~# curl localhost/api/v1/snapshots | jq  
.  
. .  
{"path": "/bp2/snapshots/site_daily_2023-02-21T01:00:03Z.snap", "name": "site_daily_2023-02-21T01:00:03Z", "solution": "site", "solution_type": "site", "label": "daily", "timestamp": "2023-02-21T01:00:03Z", "files": [".", "./api-crinoid-0", "./backup-service-0", "./bpocore-0", "./bpopg-0", "./fccustartifacts-0", "./gcs-0", "./kafka-0", "./open-api-0", "./postgres-0"]}
```

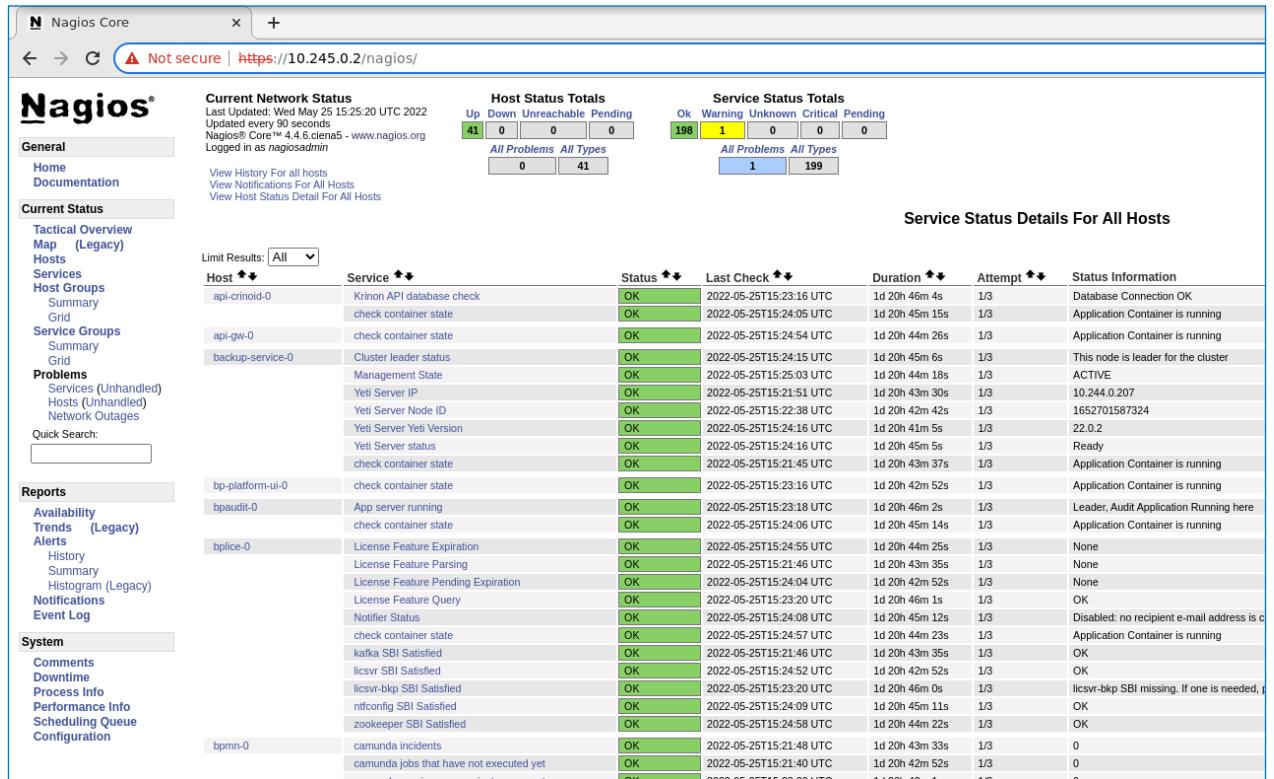
20. Exit the NSM container.

```
root@nsm-r25rd:~# exit  
$
```

Task 4: Monitoring the Blue Planet Platform with Nagios

In this task, you will use Nagios to monitor the Blue Planet Platform.

- From your desktop open a browser session to the Nagios user interface at <https://bp.lab.local/nagios>. Use the credentials **admin/adminpw** to log in. On the left side, you will find the menu options for different Nagios features, including configuration. At the top of the page is the summary status, and the majority of the window is occupied by application and service details.



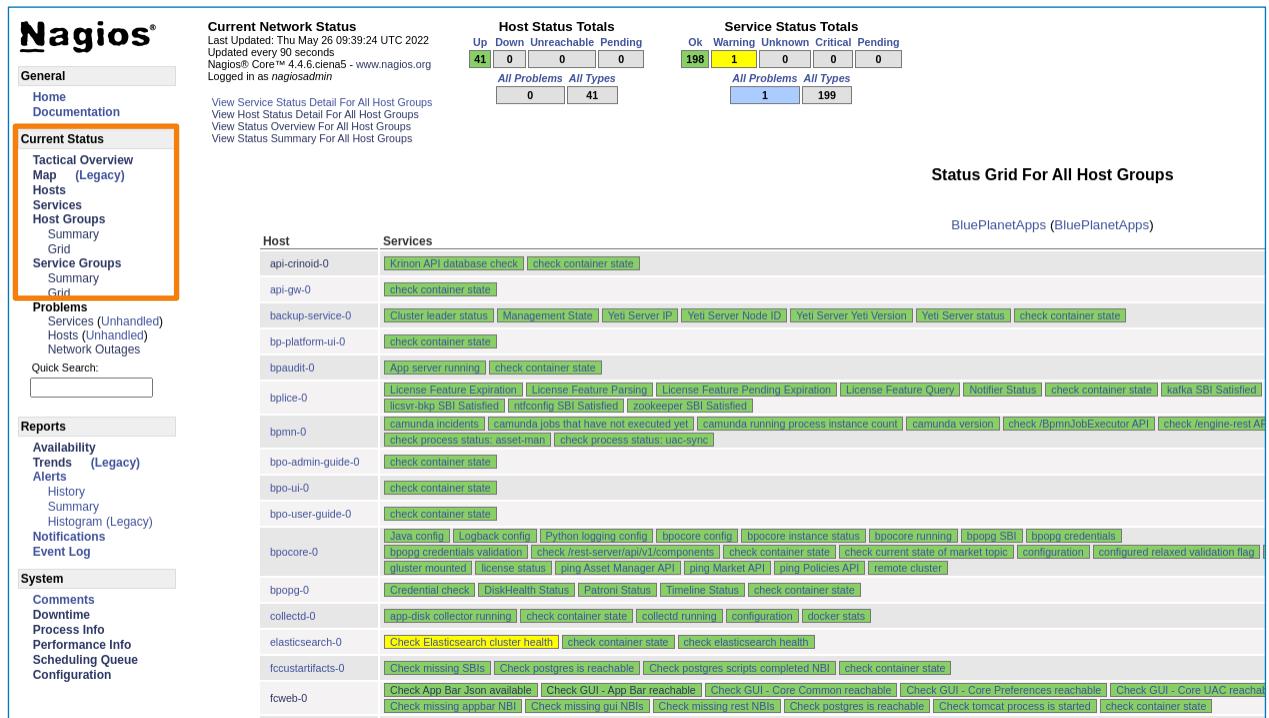
The screenshot shows the Nagios Core interface. At the top, there's a header bar with the Nagios logo and a link to <https://10.245.0.2/nagios/>. Below the header, the main content area is divided into several sections:

- Current Network Status:** Shows last updated time (Wed May 25 15:25:20 UTC 2022), Nagios Core version (4.4.6.ciena5), and a log message about being updated every 90 seconds.
- Host Status Totals:** Shows 41 hosts Up, 0 Down, 0 Unreachable, and 0 Pending. Below this are two buttons: "All Problems" and "All Types". Under "All Problems", it shows 0 and 41. Under "All Types", it shows 0 and 41.
- Service Status Totals:** Shows 198 Ok, 1 Warning, 0 Unknown, 0 Critical, and 0 Pending. Below this are two buttons: "All Problems" and "All Types". Under "All Problems", it shows 1 and 199. Under "All Types", it shows 1 and 199.
- Service Status Details For All Hosts:** A table listing various hosts and their services with their status, last check time, duration, attempt count, and status information. Some entries include notes like "Application Container is running" or "Database Connection OK".
- Left Sidebar:** Contains sections for General (Home, Documentation), Current Status (Tactical Overview, Map (Legacy), Hosts, Services, Host Groups, Service Groups, Problems, Reports, System), and Reports (Availability, Trends (Legacy), Alerts, History, Summary, Histogram (Legacy), Notifications, Event Log).

- Note the host and service status totals on the top of the page. This gives you a quick overview of the state of your server.

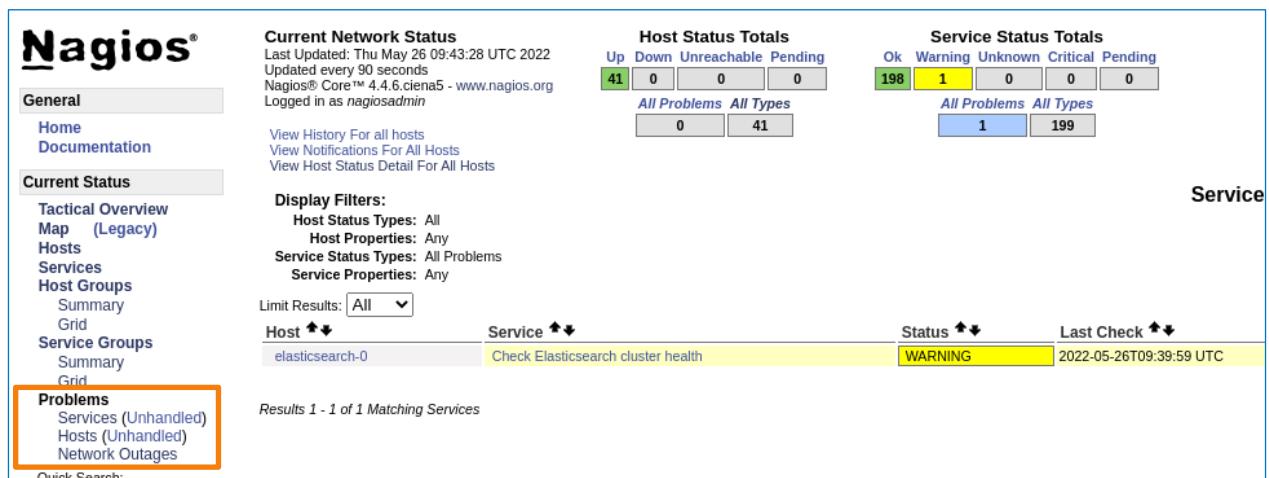


3. For different views on the hosts and services, use the options from the **Current Status** menu. For example, to display the grid-type layout for the **Host Groups**, click the **Grid** option. Optionally, take a minute to try out other options from the Current Status menu.



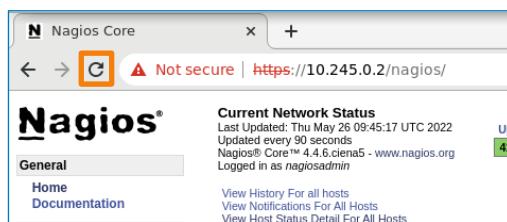
Host	Services
api-crinoid-0	[Kronion API database check] [check container state]
api-gw-0	[check container state]
backup-service-0	[Cluster leader status] [Management State] [Yeti Server IP] [Yeti Server Node ID] [Yeti Server Yeti Version] [Yeti Server status] [check container state]
bp-platform-ui-0	[check container state]
bpaudit-0	[App server running] [check container state]
bplice-0	[License Feature Expansion] [License Feature Parsing] [License Feature Pending Expiration] [License Feature Query] [Notifier Status] [check container state] [Redis SBI Satisfied] [Ircsv-bpg SBI Satisfied] [Rtconfig SBI Satisfied] [Zookeeper SBI Satisfied]
bpmm-0	[camunda incidents] [camunda jobs that have not executed yet] [camunda running process instance count] [camunda version] [check /BpmnJobExecutor API] [check /engine-rest API] [check process status: asset-man] [check process status: uac-sync]
bpo-admin-guide-0	[check container state]
bpo-ui-0	[check container state]
bpo-user-guide-0	[check container state]
bpcore-0	[Java config] [Logback config] [Python logging config] [bpcore config] [bpcore instance status] [bpcore running] [bpogg SBI] [bpogg credentials] [bpogg credentials validation] [check /rest-server/api/v1/components] [check container state] [check current state of market topic] [configuration] [configured relaxed validation flag] [Gluster mounted] [license status] [ping Asset Manager API] [ping Market API] [ping Policies API] [remote cluster]
bpogg-0	[Credential check] [DiskHealth Status] [Patroni Status] [Timeline Status] [check container state]
collectd-0	[app-disk collector running] [check container state] [collected running] [configuration] [docker stats]
elasticsearch-0	[Check Elasticsearch cluster health] [check container state] [check elasticsearch health]
fccustartifacts-0	[Check missing SBIs] [Check postgres is reachable] [Check postgres scripts completed NBI] [check container state]
fcweb-0	[Check App Bar Json available] [Check GUI - App Bar reachable] [Check GUI - Core Common reachable] [Check GUI - Core Preferences reachable] [Check GUI - Core UAC reachable] [Check missing appbar NBI] [Check missing gui NBIs] [Check missing rest NBIs] [Check postgres is reachable] [Check tomcat process is started] [check container state]

4. To filter the hosts or services that experience problems, use the options under **Current Status > Problems** menu. Click the **Services (unhandled)** option to focus on the problematic services.

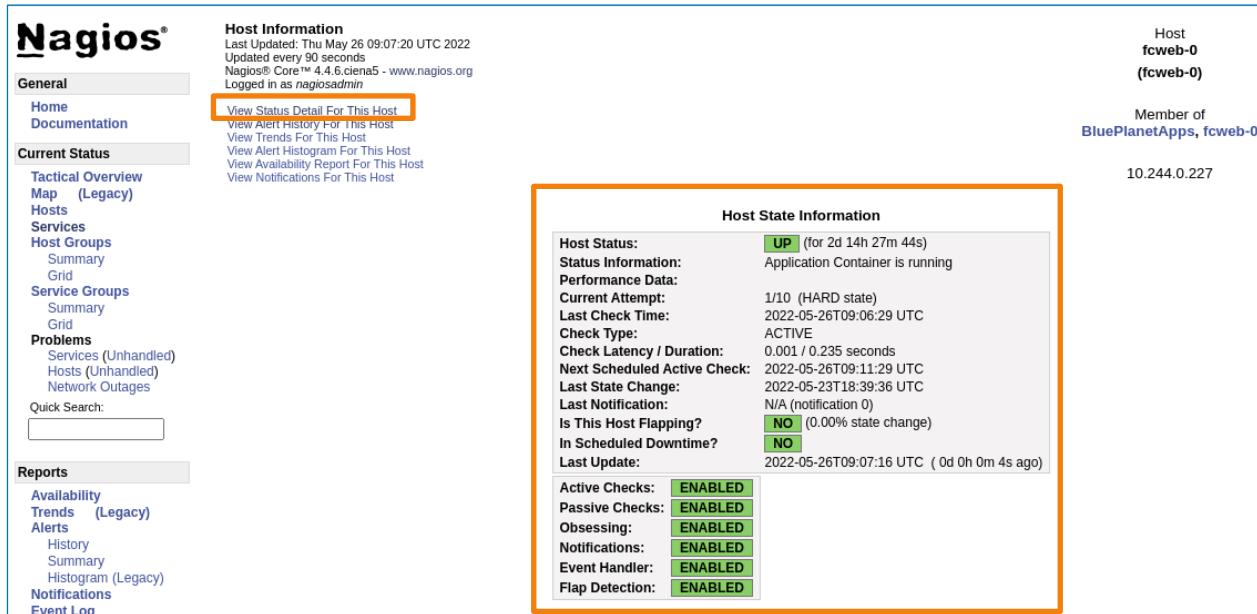


Host	Service	Status	Last Check
elasticsearch-0	Check Elasticsearch cluster health	WARNING	2022-05-26T09:39:59 UTC

5. Click the **refresh** button in your browser to bring the default view of hosts and services back.



6. The host section corresponds to Kubernetes pods and services represent Nagios checks that are executed against the pod. From the list of hosts, find **fcweb-0** and click it. The host information page will open. Observe that the container is running without problems.



Nagios®

General

- Home
- Documentation

Current Status

- Tactical Overview Map (Legacy)
- Hosts
- Services
- Host Groups
 - Summary
 - Grid
- Service Groups
 - Summary
 - Grid
- Problems
 - Services (Unhandled)
 - Hosts (Unhandled)
 - Network Outages

Quick Search:

Reports

- Availability
- Trends (Legacy)
- Alerts
 - History
 - Summary
 - Histogram (Legacy)
- Notifications
- Event Log

Host Information

Last Updated: Thu May 26 09:07:20 UTC 2022
 Updated every 90 seconds
 Nagios® Core™ 4.4.6.ciena5 - www.nagios.org
 Logged in as *nagiosadmin*

View Status Detail For This Host

Host State Information

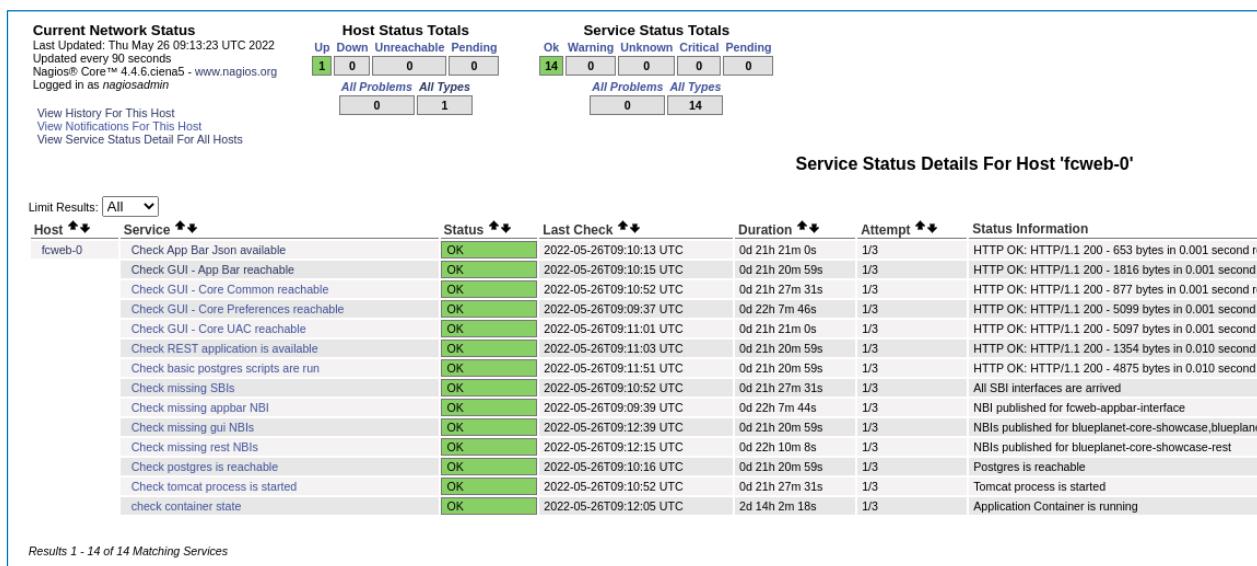
Host Status:	UP (for 2d 14h 27m 44s)
Status Information:	Application Container is running
Performance Data:	
Current Attempt:	1/10 (HARD state)
Last Check Time:	2022-05-26T09:06:29 UTC
Check Type:	ACTIVE
Check Latency / Duration:	0.001 / 0.235 seconds
Next Scheduled Active Check:	2022-05-26T09:11:29 UTC
Last State Change:	2022-05-23T18:39:36 UTC
Last Notification:	N/A (notification 0)
Is This Host Flapping?	NO (0.00% state change)
In Scheduled Downtime?	NO
Last Update:	2022-05-26T09:07:16 UTC (0d 0h 0m 4s ago)
Active Checks:	ENABLED
Passive Checks:	ENABLED
Obsessing:	ENABLED
Notifications:	ENABLED
Event Handler:	ENABLED
Flap Detection:	ENABLED

Host fcweb-0 (fcweb-0)

Member of **BluePlanetApps, fcweb-0**

10.244.0.227

7. Now click the **View Status Detail For This Host** link in the top left of the window to focus on the fcweb-0 pod.



Current Network Status

Last Updated: Thu May 26 09:13:23 UTC 2022
 Updated every 90 seconds
 Nagios® Core™ 4.4.6.ciena5 - www.nagios.org
 Logged in as *nagiosadmin*

Host Status Totals

Up	Down	Unreachable	Pending
1	0	0	0
All Problems	All Types		
0	1		

Service Status Totals

Ok	Warning	Unknown	Critical	Pending
14	0	0	0	0
All Problems	All Types			
0	14			

Service Status Details For Host 'fcweb-0'

Host	Service	Status	Last Check	Duration	Attempt	Status Information
fcweb-0	Check App Bar Json available	OK	2022-05-26T09:10:13 UTC	0d 21h 21m 0s	1/3	HTTP OK: HTTP/1.1 200 - 653 bytes in 0.001 second re
fcweb-0	Check GUI - App Bar reachable	OK	2022-05-26T09:10:15 UTC	0d 21h 20m 59s	1/3	HTTP OK: HTTP/1.1 200 - 1816 bytes in 0.001 second re
fcweb-0	Check GUI - Core Common reachable	OK	2022-05-26T09:10:52 UTC	0d 21h 27m 31s	1/3	HTTP OK: HTTP/1.1 200 - 877 bytes in 0.001 second re
fcweb-0	Check GUI - Core Preferences reachable	OK	2022-05-26T09:09:37 UTC	0d 22h 7m 46s	1/3	HTTP OK: HTTP/1.1 200 - 5099 bytes in 0.001 second re
fcweb-0	Check GUI - Core UAC reachable	OK	2022-05-26T09:11:01 UTC	0d 21h 21m 0s	1/3	HTTP OK: HTTP/1.1 200 - 5097 bytes in 0.001 second re
fcweb-0	Check REST application is available	OK	2022-05-26T09:11:03 UTC	0d 21h 20m 59s	1/3	HTTP OK: HTTP/1.1 200 - 1354 bytes in 0.010 second re
fcweb-0	Check basic postgres scripts are run	OK	2022-05-26T09:11:51 UTC	0d 21h 20m 59s	1/3	HTTP OK: HTTP/1.1 200 - 4875 bytes in 0.010 second re
fcweb-0	Check missing SBLs	OK	2022-05-26T09:10:52 UTC	0d 21h 27m 31s	1/3	All SBI interfaces are arrived
fcweb-0	Check missing appbar NBI	OK	2022-05-26T09:09:39 UTC	0d 22h 7m 44s	1/3	NBI published for fcweb-appbar-interface
fcweb-0	Check missing gui NBIs	OK	2022-05-26T09:12:39 UTC	0d 21h 20m 59s	1/3	NBIs published for blueplanet-core-showcase,blueplanet
fcweb-0	Check missing rest NBIs	OK	2022-05-26T09:12:15 UTC	0d 22h 10m 8s	1/3	NBIs published for blueplanet-core-showcase-rest
fcweb-0	Check postgres is reachable	OK	2022-05-26T09:10:16 UTC	0d 21h 20m 59s	1/3	Postgres is reachable
fcweb-0	Check tomcat process is started	OK	2022-05-26T09:10:52 UTC	0d 21h 27m 31s	1/3	Tomcat process is started
fcweb-0	check container state	OK	2022-05-26T09:12:05 UTC	2d 14h 2m 18s	1/3	Application Container is running

Results 1 - 14 of 14 Matching Services

8. Now you will see how the Nagios page will change when the pod resets. From your desktop, open a terminal window and restart the **fcweb-0** pod using the **kubectl** command.

```
[bpadmin@bp ~]$ kubectl delete pod -n soo fcweb-0 --force
warning: Immediate deletion does not wait for confirmation that the
running resource has been terminated. The resource may continue to
run on the cluster indefinitely.

pod "fcweb-0" force deleted
```

9. Switch back to the Nagios browser session and observe the changes in the fcweb-0 status. Observe how probes start to fail as they are executed.

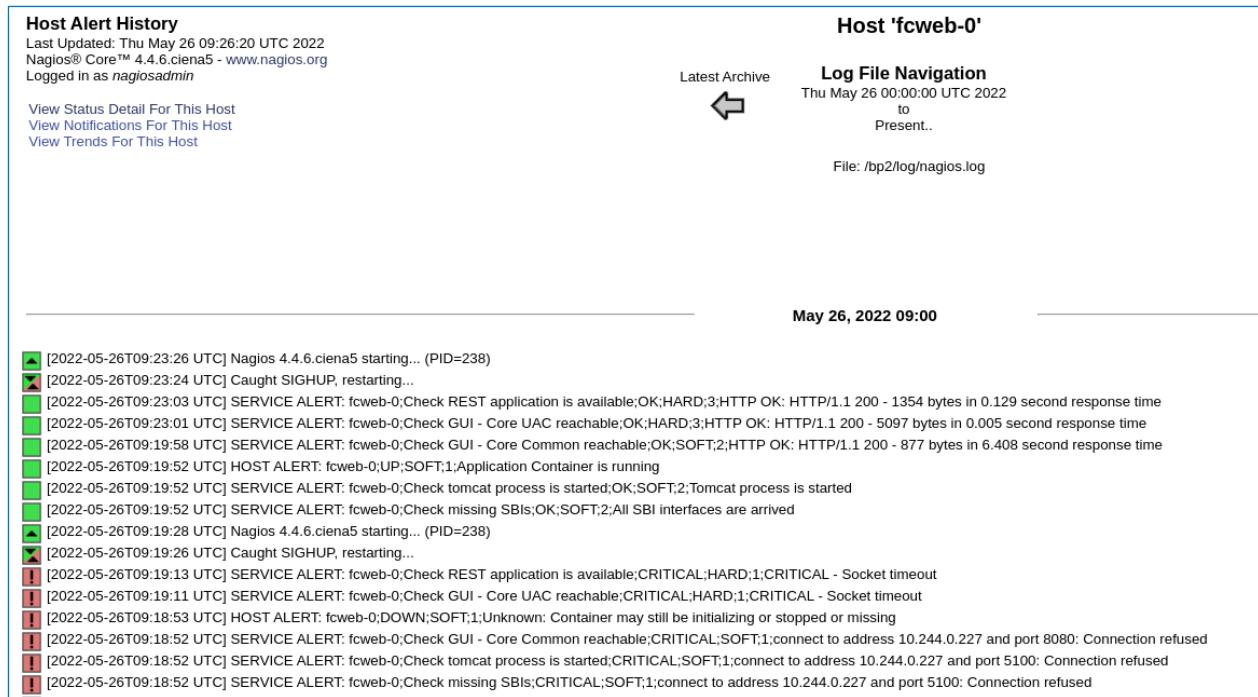
Current Network Status		Host Status Totals				Service Status Totals				
Last Updated: Thu May 26 09:19:23 UTC 2022	Updated every 90 seconds	Up	Down	Unreachable	Pending	Ok	Warning	Unknown	Critical	Pending
Nagios® Core™ 4.4.6.ciena5 - www.nagios.org		0	1	0	0	9	0	0	5	0
Logged in as nagiosadmin		All Problems		All Types		All Problems		All Types		
		1		1		5		14		
View History For This Host View Notifications For This Host View Service Status Detail For All Hosts										
Service Status Details For Host 'fcweb-0'										
Limit Results: All		Host ▲▼		Service ▲▼		Status ▲▼	Last Check ▲▼	Duration ▲▼	Attempt ▲▼	Status Information
fcweb-0						OK	2022-05-26T09:18:13 UTC	0d 21h 27m 0s	1/3	HTTP OK: HTTP/1.1 200 - 653 bytes in 0.001 second response time
						OK	2022-05-26T09:18:15 UTC	0d 21h 26m 59s	1/3	HTTP OK: HTTP/1.1 200 - 1816 bytes in 0.001 second response time
						CRITICAL	2022-05-26T09:18:52 UTC	0d 0h 0m 31s	1/3	connect to address 10.244.0.227 and port 8080: Connection refused
						OK	2022-05-26T09:17:37 UTC	0d 22h 13m 46s	1/3	HTTP OK: HTTP/1.1 200 - 5099 bytes in 0.001 second response time
						CRITICAL	2022-05-26T09:19:01 UTC	0d 0h 0m 22s	1/3	CRITICAL - Socket timeout
						CRITICAL	2022-05-26T09:19:01 UTC	0d 0h 0m 20s	1/3	CRITICAL - Socket timeout
						OK	2022-05-26T09:15:51 UTC	0d 21h 26m 59s	1/3	HTTP OK: HTTP/1.1 200 - 4875 bytes in 0.017 second response time
						CRITICAL	2022-05-26T09:18:52 UTC	0d 0h 0m 31s	1/3	connect to address 10.244.0.227 and port 5100: Connection refused
						OK	2022-05-26T09:17:39 UTC	0d 22h 13m 44s	1/3	NBI published for fcweb-appbar-interface
						OK	2022-05-26T09:16:39 UTC	0d 21h 26m 59s	1/3	NBIs published for blueplanet-core-showcase_blueplanet-app-bar-ui_bluep
						OK	2022-05-26T09:16:15 UTC	0d 22h 16m 8s	1/3	NBIs published for blueplanet-core-showcase-rest
						OK	2022-05-26T09:18:16 UTC	0d 21h 26m 59s	1/3	Postgres is reachable
						CRITICAL	2022-05-26T09:18:52 UTC	0d 0h 0m 31s	1/3	connect to address 10.244.0.227 and port 5100: Connection refused
						OK	2022-05-26T09:16:05 UTC	2d 14h 8m 18s	1/3	Application Container is running

NOTE: You will have to wait a short refresh period to observe the changes.

10. After a while, all the services will turn back green as the pod redeploys and transitions into the Running state. Now click the **View History For This Host** link from the top left of the window to get a history of events for the reset procedure.

Current Network Status		Host Status Totals				Service Status Totals				
Last Updated: Thu May 26 09:23:35 UTC 2022	Updated every 90 seconds	Up	Down	Unreachable	Pending	Ok	Warning	Unknown	Critical	Pending
Nagios® Core™ 4.4.6.ciena5 - www.nagios.org		1	0	0	0	14	0	0	0	0
Logged in as nagiosadmin		All Problems		All Types		All Problems		All Types		
		0		1		0		14		
View History For This Host View Notifications For This Host View Service Status Detail For All Hosts										
Serv										
Limit Results: All		Host ▲▼		Service ▲▼		Status ▲▼	Last Check ▲▼	Duration ▲▼		
fcweb-0						OK	2022-05-26T09:22:13 UTC	0d 21h 31m 12s		
						OK	2022-05-26T09:22:15 UTC	0d 21h 31m 11s		
						OK	2022-05-26T09:20:58 UTC	0d 0h 3m 43s		
						OK	2022-05-26T09:21:37 UTC	0d 22h 17m 58s		
						OK	2022-05-26T09:23:01 UTC	0d 0h 0m 34s		
						OK	2022-05-26T09:23:03 UTC	0d 0h 0m 32s		
						OK	2022-05-26T09:19:51 UTC	0d 21h 31m 11s		
						OK	2022-05-26T09:20:52 UTC	0d 0h 3m 43s		
						OK	2022-05-26T09:21:39 UTC	0d 22h 17m 56s		
						OK	2022-05-26T09:20:39 UTC	0d 21h 31m 11s		
						OK	2022-05-26T09:20:15 UTC	0d 22h 20m 20s		

11. Note that service alerts started at the time you reset the pod, as denoted by the red icons, which later transitioned to green when the probes got positive results.



Host Alert History

Last Updated: Thu May 26 09:26:20 UTC 2022
 Nagios® Core™ 4.4.6.ciena5 - www.nagios.org
 Logged in as nagiosadmin

[View Status Detail For This Host](#)
[View Notifications For This Host](#)
[View Trends For This Host](#)

Host 'fcweb-0'

Latest Archive 

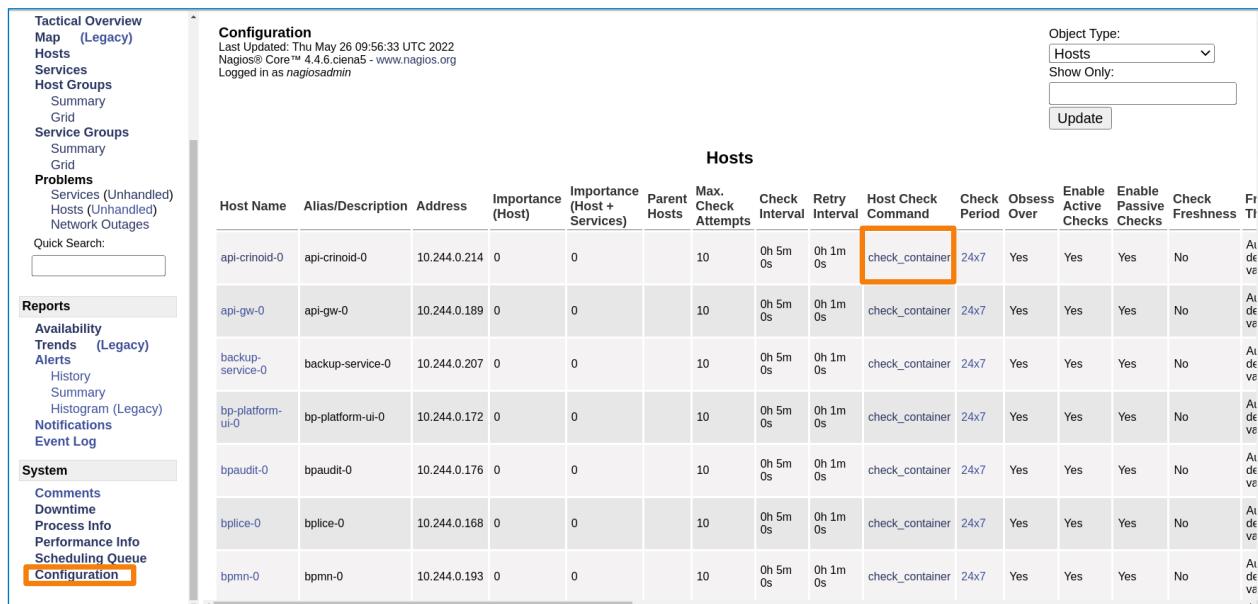
Log File Navigation

Thu May 26 00:00:00 UTC 2022 to Present..
 File: /bp2/log/nagios.log

May 26, 2022 09:00

[2022-05-26T09:23:26 UTC] Nagios 4.4.6.ciena5 starting... (PID=238)
 [2022-05-26T09:23:24 UTC] Caught SIGHUP, restarting...
 [2022-05-26T09:23:03 UTC] SERVICE ALERT: fcweb-0;Check REST application is available;OK;HARD;3;HTTP OK: HTTP/1.1 200 - 1354 bytes in 0.129 second response time
 [2022-05-26T09:23:01 UTC] SERVICE ALERT: fcweb-0;Check GUI - Core UAC reachable;OK;HARD;3;HTTP OK: HTTP/1.1 200 - 5097 bytes in 0.005 second response time
 [2022-05-26T09:19:58 UTC] SERVICE ALERT: fcweb-0;Check GUI - Core Common reachable;OK;SOFT;2;HTTP OK: HTTP/1.1 200 - 877 bytes in 6.408 second response time
 [2022-05-26T09:19:52 UTC] HOST ALERT: fcweb-0;UP;SOFT;1;Application Container is running
 [2022-05-26T09:19:52 UTC] SERVICE ALERT: fcweb-0;Check tomcat process is started;OK;SOFT;2;Tomcat process is started
 [2022-05-26T09:19:52 UTC] SERVICE ALERT: fcweb-0;Check missing SBIs;OK;SOFT;2;All SBI Interfaces are arrived
 [2022-05-26T09:19:28 UTC] Nagios 4.4.6.ciena5 starting... (PID=238)
 [2022-05-26T09:19:26 UTC] Caught SIGHUP, restarting...
 [2022-05-26T09:19:13 UTC] SERVICE ALERT: fcweb-0;Check REST application is available;CRITICAL;HARD;1;CRITICAL - Socket timeout
 [2022-05-26T09:19:11 UTC] SERVICE ALERT: fcweb-0;Check GUI - Core UAC reachable;CRITICAL;HARD;1;CRITICAL - Socket timeout
 [2022-05-26T09:18:53 UTC] HOST ALERT: fcweb-0;DOWN;SOFT;1;Unknown: Container may still be initializing or stopped or missing
 [2022-05-26T09:18:52 UTC] SERVICE ALERT: fcweb-0;Check GUI - Core Common reachable;CRITICAL;SOFT;1;connect to address 10.244.0.227 and port 8080: Connection refused
 [2022-05-26T09:18:52 UTC] SERVICE ALERT: fcweb-0;Check tomcat process is started;CRITICAL;SOFT;1;connect to address 10.244.0.227 and port 5100: Connection refused
 [2022-05-26T09:18:52 UTC] SERVICE ALERT: fcweb-0;Check missing SBIs;CRITICAL;SOFT;1;connect to address 10.244.0.227 and port 5100: Connection refused

12. Now, check the configuration of hosts and services by clicking the **Configuration** option under the **System** menu. For Object Type, choose **Hosts** and click **Continue**. The list of hosts opens with the respective host check commands.



Tactical Overview

[Map \(Legacy\)](#)
[Hosts](#)
[Services](#)
[Host Groups](#)
[Summary](#)
[Grid](#)
[Service Groups](#)
[Summary](#)
[Grid](#)
Problems
[Services \(Unhandled\)](#)
[Hosts \(Unhandled\)](#)
[Network Outages](#)
Quick Search:

Reports

[Availability Trends \(Legacy\)](#)
[Alerts History](#)
[Summary](#)
[Histogram \(Legacy\)](#)
[Notifications](#)
[Event Log](#)

System

[Comments](#)
[Downtime](#)
[Process Info](#)
[Performance Info](#)
[Scheduling Queue](#)
Configuration

Configuration

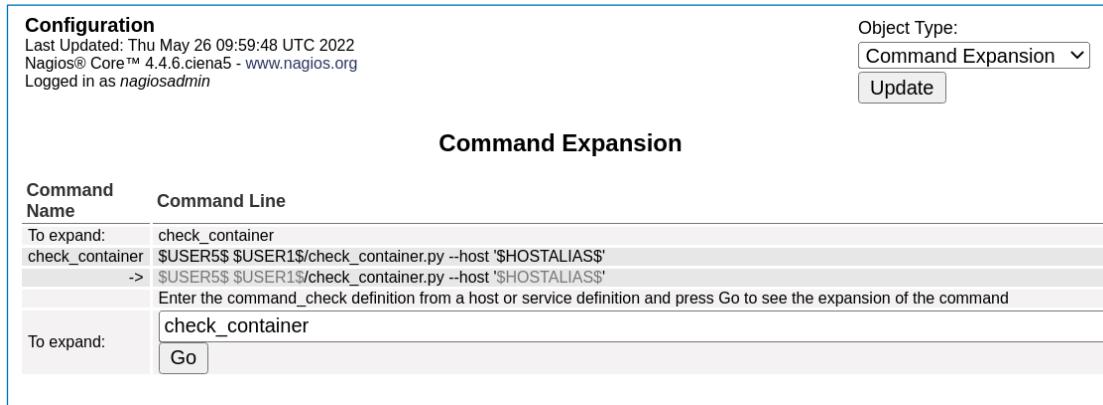
Last Updated: Thu May 26 09:58:33 UTC 2022
 Nagios® Core™ 4.4.6.ciena5 - www.nagios.org
 Logged in as nagiosadmin

Object Type: **Hosts**

Hosts

Host Name	Alias/Description	Address	Importance (Host)	Importance (Host + Services)	Parent Hosts	Max. Check Attempts	Check Interval	Retry Interval	Host Check Command	Check Period	Obsess Over	Enable Active Checks	Enable Passive Checks	Check Freshness	File
api-crinoid-0	api-crinoid-0	10.244.0.214	0	0		10	0h 5m 0s	0h 1m 0s	check_container	24x7	Yes	Yes	Yes	No	Aideve
api-gw-0	api-gw-0	10.244.0.189	0	0		10	0h 5m 0s	0h 1m 0s	check_container	24x7	Yes	Yes	Yes	No	Aideve
backup-service-0	backup-service-0	10.244.0.207	0	0		10	0h 5m 0s	0h 1m 0s	check_container	24x7	Yes	Yes	Yes	No	Aideve
bp-platform-ui-0	bp-platform-ui-0	10.244.0.172	0	0		10	0h 5m 0s	0h 1m 0s	check_container	24x7	Yes	Yes	Yes	No	Aideve
bpaudit-0	bpaudit-0	10.244.0.176	0	0		10	0h 5m 0s	0h 1m 0s	check_container	24x7	Yes	Yes	Yes	No	Aideve
bplice-0	bplice-0	10.244.0.168	0	0		10	0h 5m 0s	0h 1m 0s	check_container	24x7	Yes	Yes	Yes	No	Aideve
bpmn-0	bpmn-0	10.244.0.193	0	0		10	0h 5m 0s	0h 1m 0s	check_container	24x7	Yes	Yes	Yes	No	Aideve

13. To view the definition of the host check command, click the **check_container** hyperlink for the first table entry, **api-crinoid-0**. Note how for the container check, a Python script named **check_container.py** is executed from Nagios.

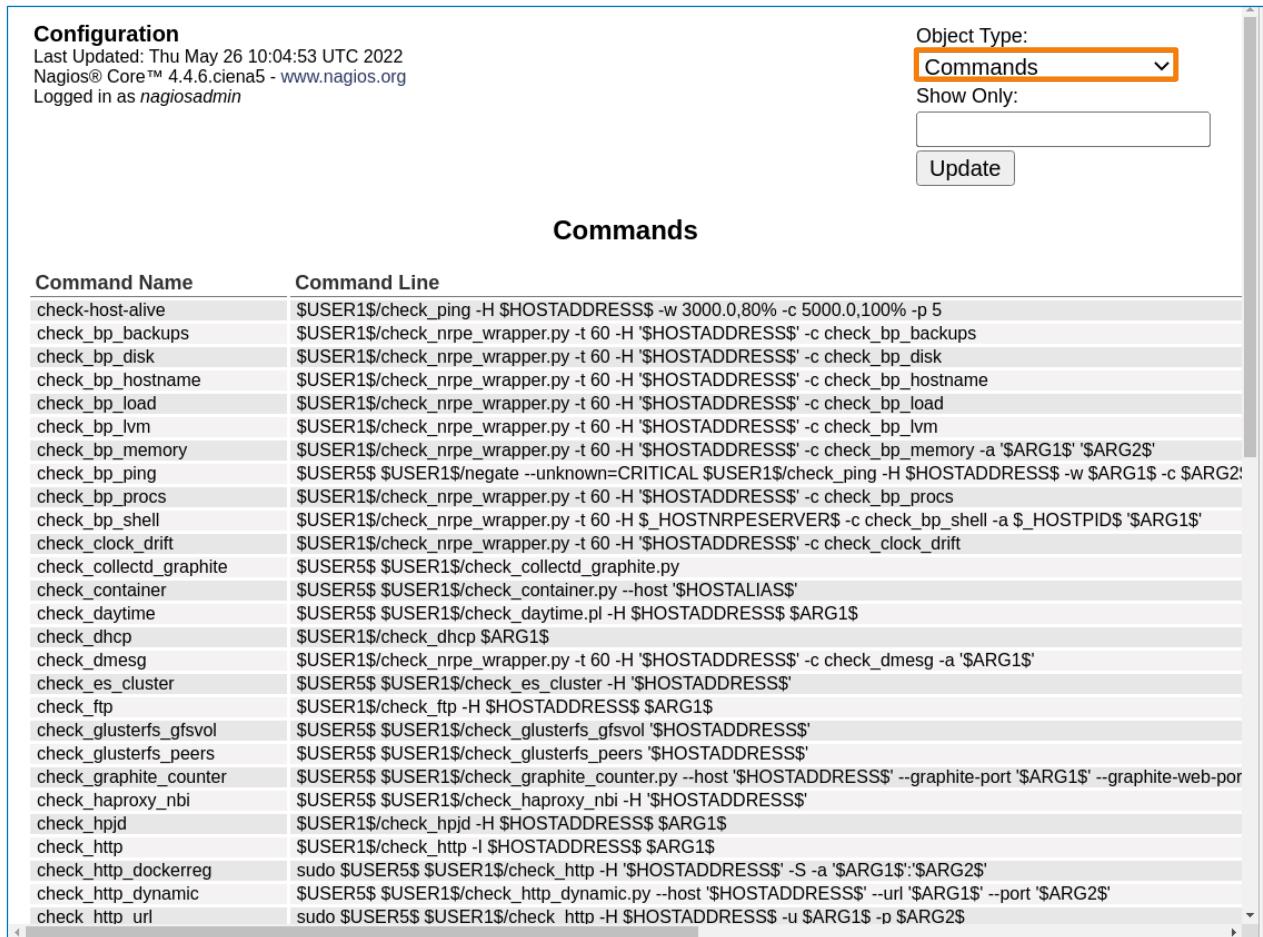


The screenshot shows the Nagios configuration interface with the following details:

- Configuration** section:
 - Last Updated: Thu May 26 09:59:48 UTC 2022
 - Nagios® Core™ 4.4.6.ciena5 - www.nagios.org
 - Logged in as *nagiosadmin*
- Object Type:** Command Expansion
- Update** button
- Command Expansion** section:

Command Name	Command Line
To expand: check_container	<code>\$USER5\$ \$USER1\$/check_container.py --host '\$HOSTALIAS\$'</code> -> <code>\$USER5\$ \$USER1\$/check_container.py --host '\$HOSTALIAS\$'</code>
Enter the command_check definition from a host or service definition and press Go to see the expansion of the command	
To expand: check_container	Go

14. To get a complete list of defined checks that can be executed for various hosts and services, under **Object Type**, choose **Commands** and click **Update**. The list of commands with the respective CLI commands is displayed.



The screenshot shows the Nagios configuration interface with the following details:

- Configuration** section:
 - Last Updated: Thu May 26 10:04:53 UTC 2022
 - Nagios® Core™ 4.4.6.ciena5 - www.nagios.org
 - Logged in as *nagiosadmin*
- Object Type:** Commands
- Show Only:** (empty input field)
- Update** button
- Commands** section:

Command Name	Command Line
check-host-alive	<code>\$USER1\$/check_ping -H \$HOSTADDRESS\$ -w 3000.0.80% -c 5000.0.100% -p 5</code>
check_bp_backups	<code>\$USER1\$/check_nrpe_wrapper.py -t 60 -H '\$HOSTADDRESS\$' -c check_bp_backups</code>
check_bp_disk	<code>\$USER1\$/check_nrpe_wrapper.py -t 60 -H '\$HOSTADDRESS\$' -c check_bp_disk</code>
check_bp_hostname	<code>\$USER1\$/check_nrpe_wrapper.py -t 60 -H '\$HOSTADDRESS\$' -c check_bp_hostname</code>
check_bp_load	<code>\$USER1\$/check_nrpe_wrapper.py -t 60 -H '\$HOSTADDRESS\$' -c check_bp_load</code>
check_bp_lvm	<code>\$USER1\$/check_nrpe_wrapper.py -t 60 -H '\$HOSTADDRESS\$' -c check_bp_lvm</code>
check_bp_memory	<code>\$USER1\$/check_nrpe_wrapper.py -t 60 -H '\$HOSTADDRESS\$' -c check_bp_memory -a '\$ARG1\$' '\$ARG2\$'</code>
check_bp_ping	<code>\$USER5\$ \$USER1\$/negate --unknown=CRITICAL \$USER1\$/check_ping -H \$HOSTADDRESS\$ -w \$ARG1\$ -c \$ARG2\$</code>
check_bp_procs	<code>\$USER1\$/check_nrpe_wrapper.py -t 60 -H '\$HOSTADDRESS\$' -c check_bp_procs</code>
check_bp_shell	<code>\$USER1\$/check_nrpe_wrapper.py -t 60 -H \$_HOSTNRPE\$ -c check_bp_shell -a \$_HOSTPID\$ '\$ARG1\$'</code>
check_clock_drift	<code>\$USER1\$/check_nrpe_wrapper.py -t 60 -H '\$HOSTADDRESS\$' -c check_clock_drift</code>
check_collectd_graphite	<code>\$USER5\$ \$USER1\$/check_collectd_graphite.py</code>
check_container	<code>\$USER5\$ \$USER1\$/check_container.py --host '\$HOSTALIAS\$'</code>
check_daytime	<code>\$USER5\$ \$USER1\$/check_daytime.pl -H \$HOSTADDRESS\$ \$ARG1\$</code>
check_dhcp	<code>\$USER1\$/check_dhcp \$ARG1\$</code>
check_dmesg	<code>\$USER1\$/check_nrpe_wrapper.py -t 60 -H '\$HOSTADDRESS\$' -c check_dmesg -a '\$ARG1\$'</code>
check_es_cluster	<code>\$USER5\$ \$USER1\$/check_es_cluster -H '\$HOSTADDRESS\$'</code>
check_ftp	<code>\$USER1\$/check_ftp -H \$HOSTADDRESS\$ \$ARG1\$</code>
check_glusterfs_gfsvol	<code>\$USER5\$ \$USER1\$/check_glusterfs_gfsvol '\$HOSTADDRESS\$'</code>
check_glusterfs_peers	<code>\$USER5\$ \$USER1\$/check_glusterfs_peers '\$HOSTADDRESS\$'</code>
check_graphite_counter	<code>\$USER5\$ \$USER1\$/check_graphite_counter.py --host '\$HOSTADDRESS\$' --graphite-port '\$ARG1\$' --graphite-web-port '\$ARG2\$'</code>
check_haproxy_nbi	<code>\$USER5\$ \$USER1\$/check_haproxy_nbi -H '\$HOSTADDRESS\$'</code>
check_hpjd	<code>\$USER1\$/check_hpjd -H \$HOSTADDRESS\$ \$ARG1\$</code>
check_http	<code>\$USER1\$/check_http -I \$HOSTADDRESS\$ \$ARG1\$</code>
check_http_dockerreg	<code>sudo \$USER5\$ \$USER1\$/check_http -H '\$HOSTADDRESS\$' -S -a '\$ARG1\$:\$ARG2\$'</code>
check_http_dynamic	<code>\$USER5\$ \$USER1\$/check_http_dynamic.py --host '\$HOSTADDRESS\$' --url '\$ARG1\$' --port '\$ARG2\$'</code>
check_http_url	<code>sudo \$USER5\$ \$USER1\$/check_http -H \$HOSTADDRESS\$ -u \$ARG1\$ -p \$ARG2\$</code>

15. Open another terminal window from your desktop and connect to the Nagios container.

```
[bpadmin@bp ~]$ kubectl exec -it -n soo nagios-0 -- bash
```

- ```
root@nagios-0:/bp2/src#
```
16. The **check\_container.py** script is located in the **/bp2/data** directory. Run a container check against the **fcweb-0** pod. The result is that the container is up and running.
- ```
root@nagios-0:/bp2/src# /bp2/data/plugins/check_container.py -H fcweb-0
Application Container is running
```
17. Now change the directory to **nagios/plugins** and list the directory. You will find all the check scripts and binaries in this location.

```
root@nagios-0:/bp2/src# cd nagios/plugins
root@nagios-0:/bp2/src/nagios/plugins# ll
total 7700
drwxr-xr-x 1 root root    25 May 23 18:37 .
drwxr-xr-x 1 root root   102 May 23 18:38 ../
drwxr-xr-x 2 root root    44 May 23 18:37 __pycache__/
-rwxr-xr-x 1 root root  2163 Jan 28 08:45 check_6500ra.py*
-rwxr-xr-x 1 root root 233304 Jan 28 08:45 check_apt*
-rwxr-xr-x 1 root root  1509 Jan 28 08:45 check_bp_memory*
-rwxr-xr-x 1 root root  2346 Jan 28 08:45 check_breeze*
lrwxrwxrwx 1 root root      9 Jan 28 08:45 check_clamd -> check_tcp*
-rwxr-xr-x 1 root root 174094 Jan 28 08:45 check_cluster*
-rwxr-xr-x 1 root root  1536 Jan 28 08:45 check_collectd_graphite.py*
-rwxr-xr-x 1 root root  2703 Jan 28 08:45 check_container.py*
-rwxr-xr-x 1 root root  9613 Jan 28 08:45 check_daytime.pl*
-rwxr-xr-x 1 root root 240287 Jan 28 08:45 check_dhcp*
-rwxr-xr-x 1 root root 244765 Jan 28 08:45 check_disk*
-rwxr-xr-x 1 root root  9469 Jan 28 08:45 check_disk_smb*
-rwxr-xr-x 1 root root 135030 Jan 28 08:45 check_dummy*
-rwxr-xr-x 1 root root  1857 Jan 28 08:45 check_es_cluster*
-rwxr-xr-x 1 root root  3860 Jan 28 08:45 check_file_age*
-rwxr-xr-x 1 root root  6504 Jan 28 08:45 check_flexlm*
lrwxrwxrwx 1 root root      9 Jan 28 08:45 check_ftp -> check_tcp*
-rwxr-xr-x 1 root root  1786 Jan 28 08:45 check_glusterfs_gfsvol*
-rwxr-xr-x 1 root root  2781 Jan 28 08:45 check_glusterfs_peers*
-rwxr-xr-x 1 root root  3167 Jan 28 08:45 check_graphite_counter.py*
-rwxr-xr-x 1 root root  1054 Jan 28 08:45 check_haproxy_nbi*
-rwxr-xr-x 1 root root 228837 Jan 28 08:45 check_hpjd*
```

...(output omitted)

18. Test the https service of your Blue Planet server by using the check_https script.

```
root@nagios-0:/bp2/src/nagios/plugins# ./check_https.py -u https://10.245.0.2  
200 -- HTTP/1.1 200 OK
```

Task 5: Checking System Performance Metrics with Grafana

Grafana is the performance metrics visualization component of the Blue Planet Platform. It is accessible through the BPP WebUI or directly through the URL provided in the deployment documentation.

You will use Grafana to check the performance of different system components like hosts, applications, networking, and so on.

Grafana will use metrics data provided by different data sources and visualize it through Panels with charts, gauges, tables, or graphs.

Different layouts and combinations of Panels can be saved as Dashboards. BPP comes with many pre-configured Dashboards, but you can also create your own custom ones.

In this task, you will learn how to use Grafana's pre-defined dashboards to check system health and how to create a custom dashboard.

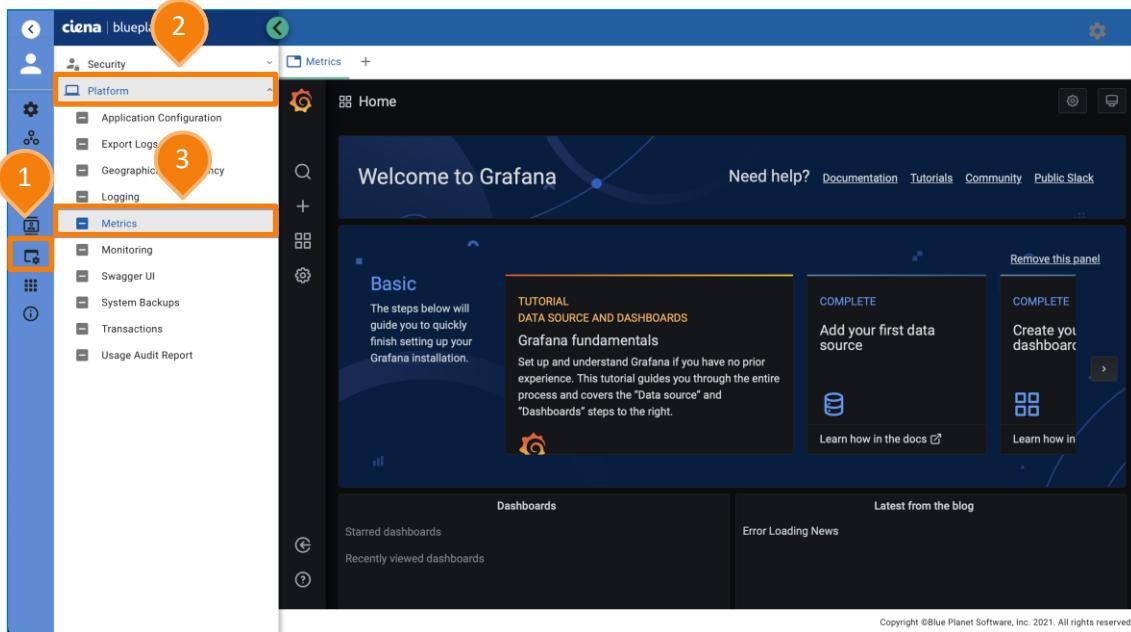
1. Log in to the Blue Planet Platform UI. From your Student VM session, open a web browser and enter **https://bp.lab.local** in the address space, and press **Enter**.
At the login screen, use the credentials **admin/adminpw** and Tenant **master**, and click the **Login** button.

The screenshot shows the login interface for the Blue Planet Platform. It includes a logo, input fields for username, password, and tenant, and links for password recovery and logging in.

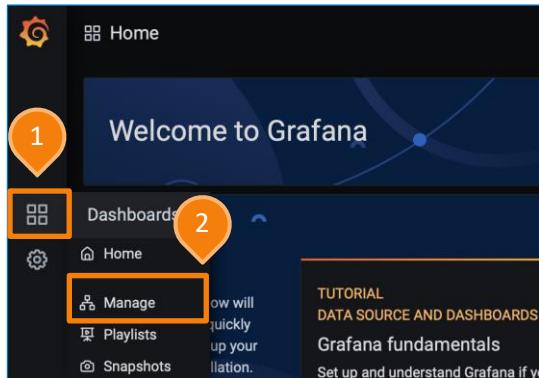
2. In the next step, you will be asked to enter a Subtenant. Leave this field blank and just click the **Apply** button.

A screenshot of a web-based configuration interface. At the top center is a blue circular logo with the letters 'bp'. Below it is a dropdown menu labeled 'Subtenant:' with a small downward arrow icon. To the right of the dropdown is a blue rectangular button with the word 'Apply' in white. The entire interface is enclosed in a light blue border.

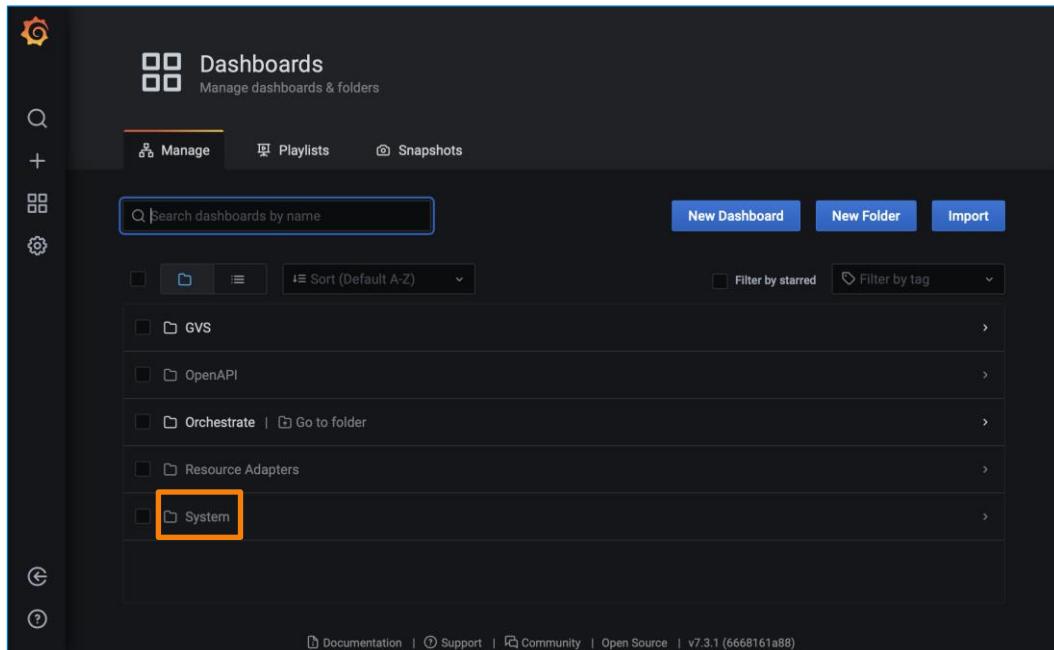
3. To access Grafana, click the **System** icon from the main menu, then click the **Platform** option and select **Metrics**.



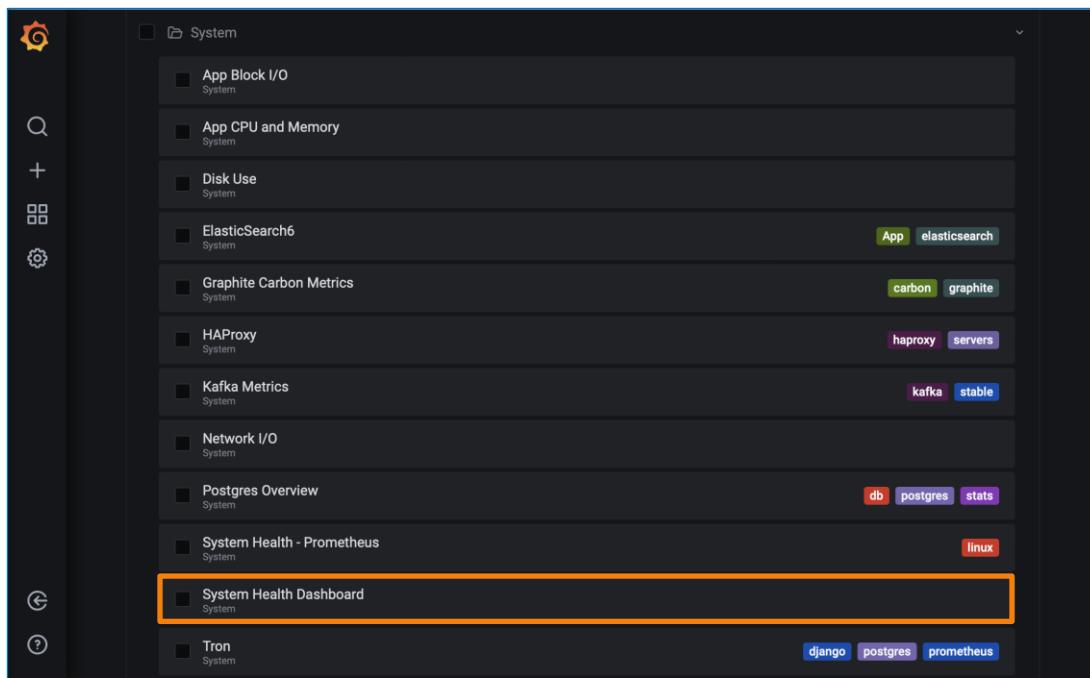
4. You will now check the system's health with pre-defined BPP dashboards. To access the list of dashboards from Grafana's menu, hover the mouse pointer on the **Dashboard** icon and click the **Manage** option.



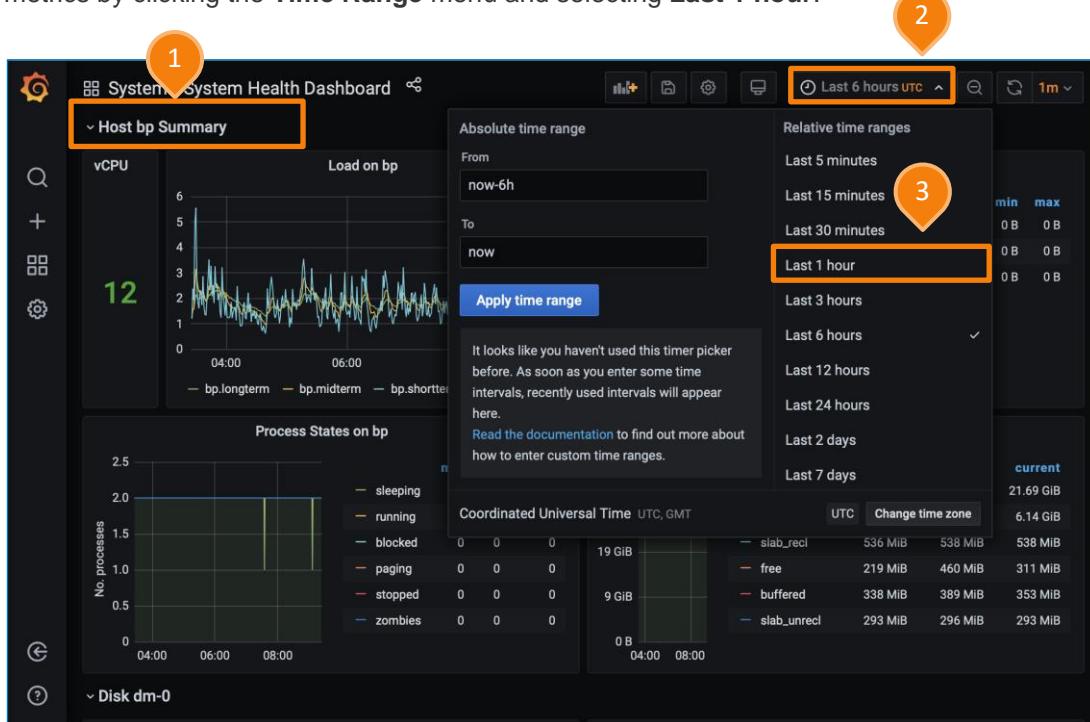
5. Click the **System** folder.



6. A list of pre-configured Dashboards for System checks will open. Click the **System Health Dashboard** option.



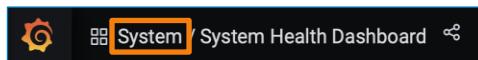
7. Scroll down to the **Host bp summary** section. You will now change the time range of displayed metrics by clicking the **Time Range** menu and selecting **Last 1 hour**.



8. Charts will now show only data from the last hour. If you want to see live data updates, change the auto-refresh interval to 5 seconds by clicking the **Refresh Interval** menu and select the **5s** option.

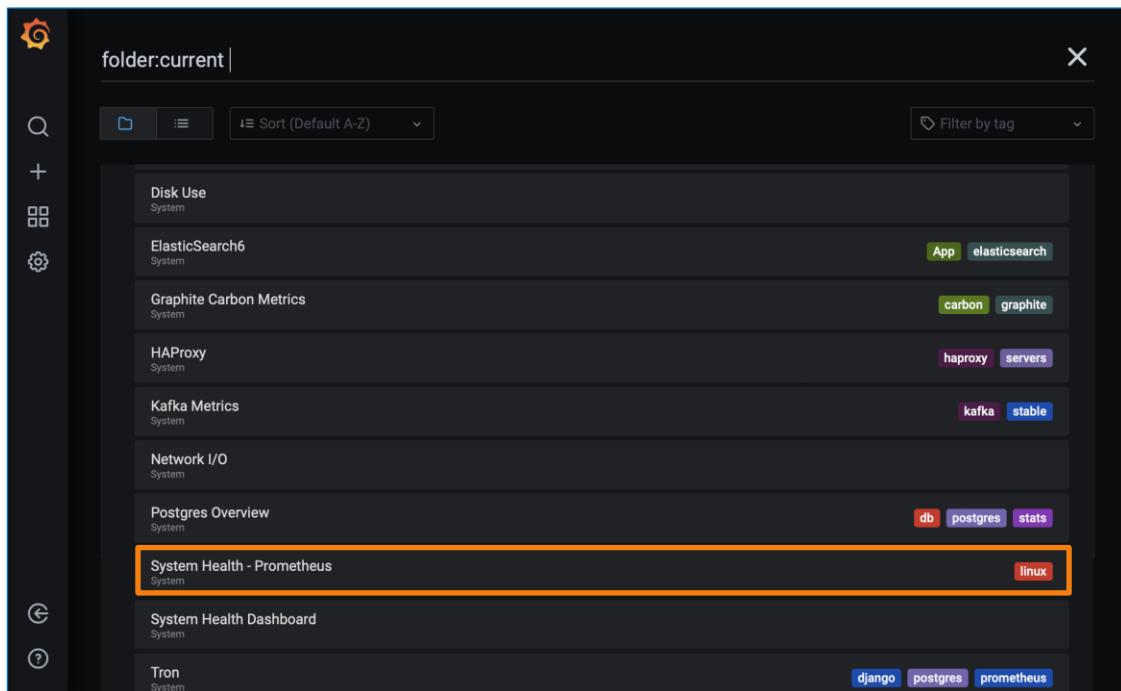


9. Next, you will check another pre-configured Dashboard with Gauges for System health checks. Go back to the System folder with Dashboards by clicking **System** in the dashboard path section.



NOTE: If you are asked to save the changes to Dashboard, click the **Discard** button.

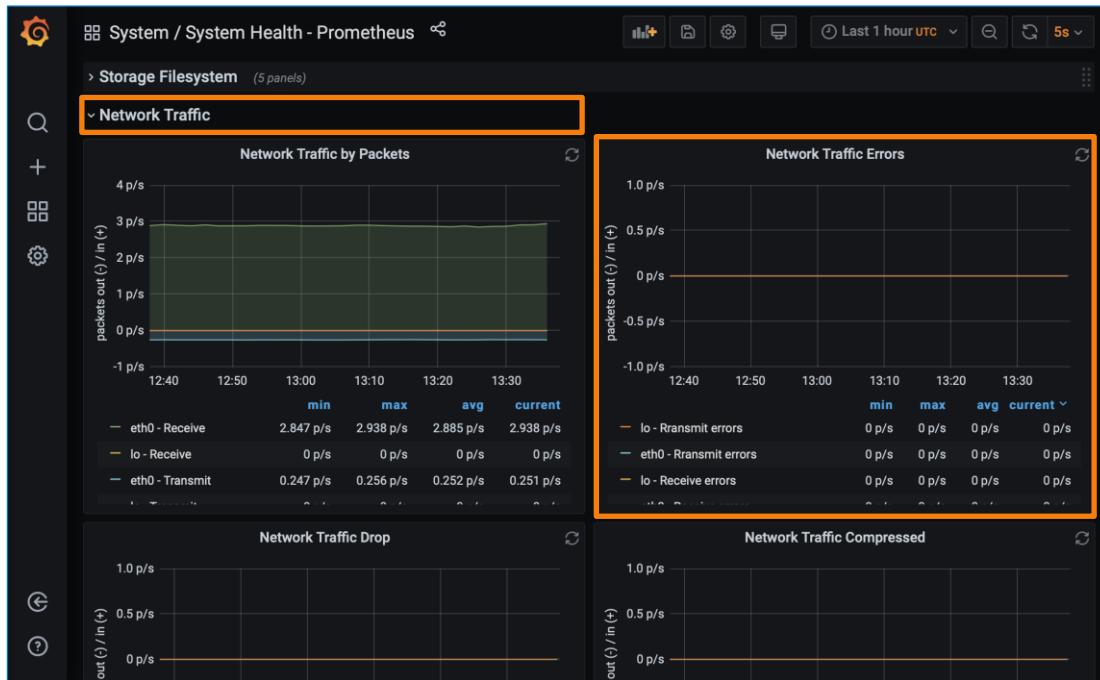
10. Scroll down and click the **System Health - Prometheus** Dashboard.



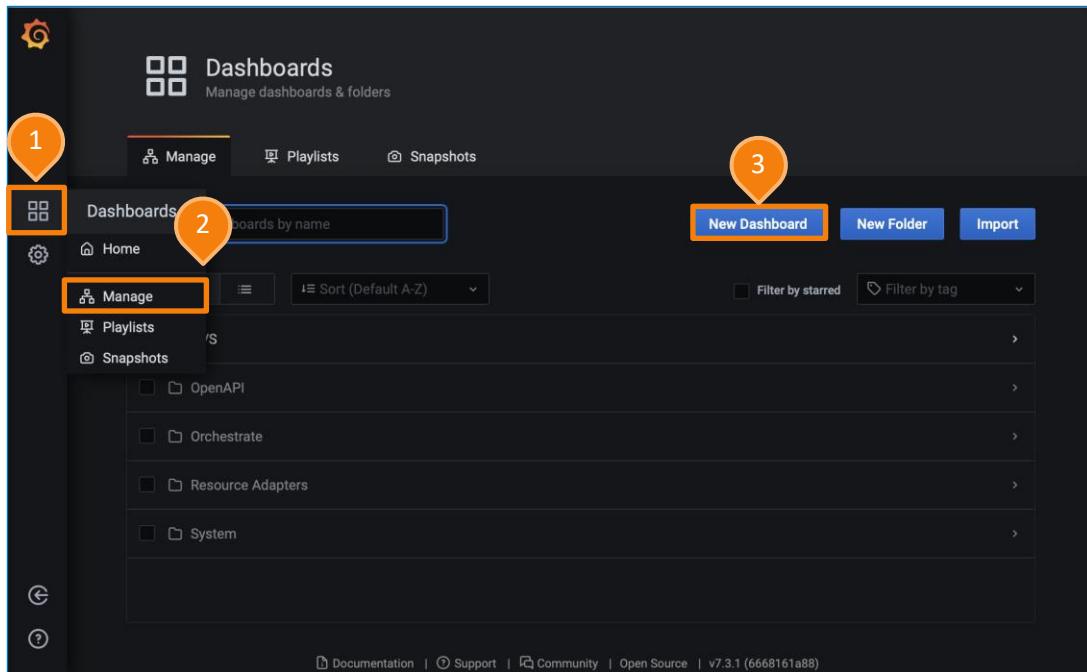
11. On this Dashboard, you can check different gauges and graphs to verify system performance. Make sure the **CPU Busy** and **RAM Used** gauges values are below 100%.



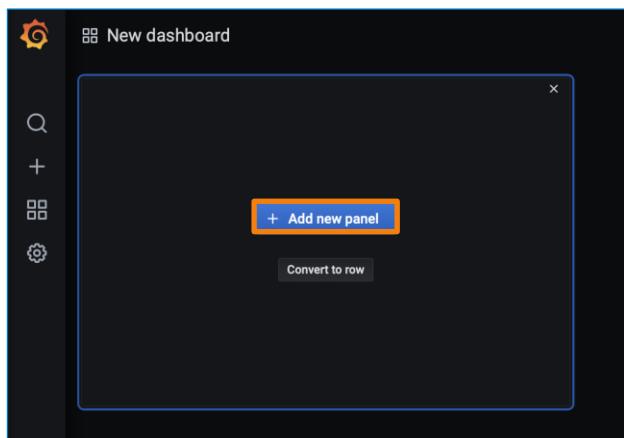
12. Scroll down and click **Network Traffic** to verify that the network is performing well and there are no **Network Traffic Errors**.



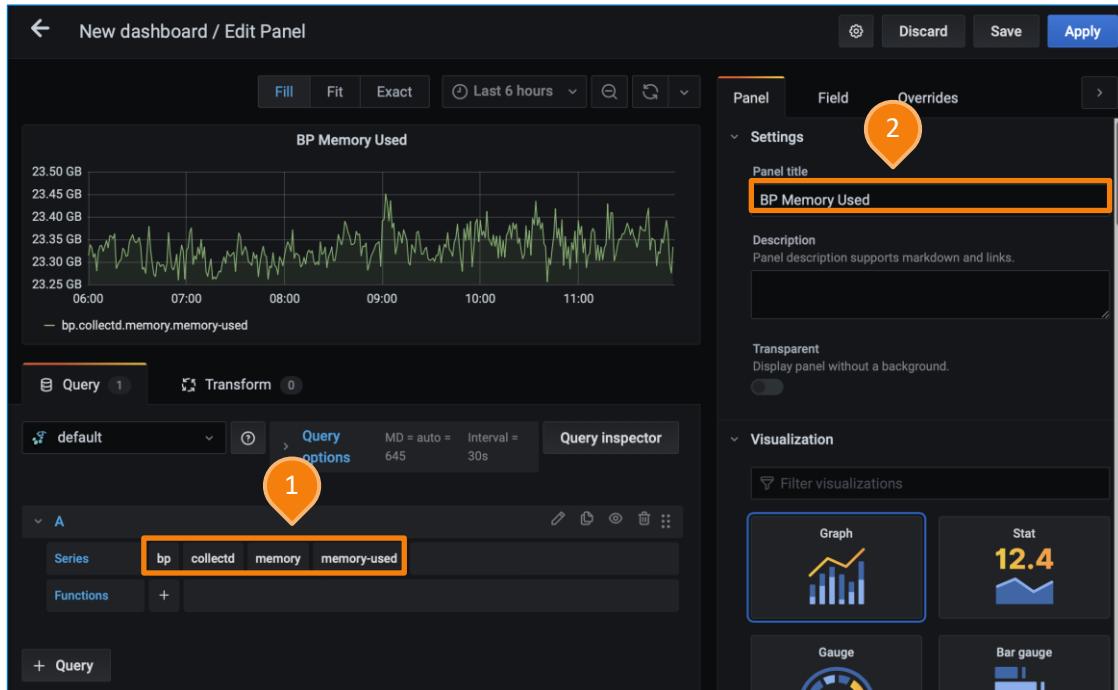
13. Next you will create a custom Dashboard. Hover the mouse pointer over the **Dashboard** icon and click the **Manage** option. Then click the **New Dashboard** button.



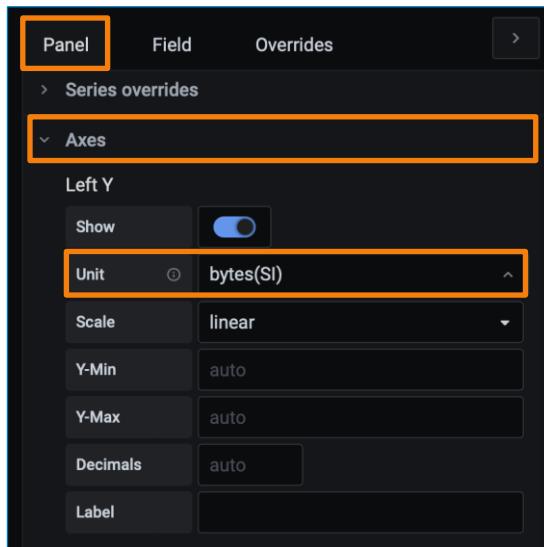
14. Dashboards consist of Panels. You will create one new Panel for the Blue Planet memory utilization chart. Click the **+ Add new panel** button.



15. First select the data you want to see on the chart by clicking on the fields in the **Series** row and selecting: **bp – collectd – memory – memory-used**. Next, write the Panel Title **BP Memory Used**.



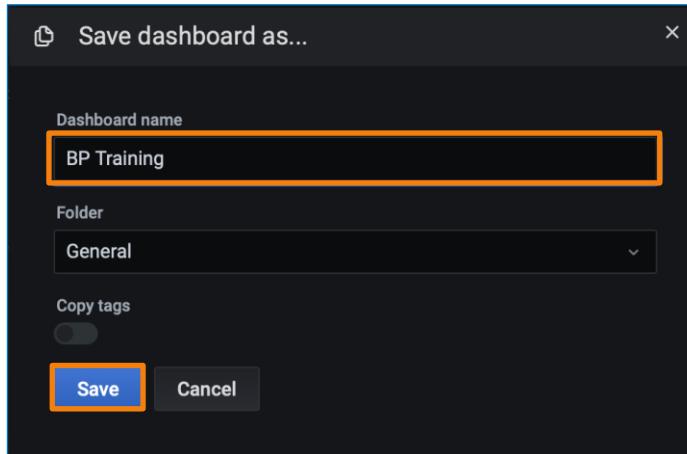
16. Scroll down the **Panel** frame and click the **Axes** row. For the **Left Y axis**, select **Unit = bytes(SI)** to show memory usage on the chart in bytes. Bytes(SI) option is available under the **Data** group.



17. Click the **Save** button in the top right corner to save and apply Dashboard and Panel changes.



18. You will be asked to give the Dashboard a name. Enter **BP Training** and click the **Save** button.



19. A green message box should appear to confirm that the Dashboard was successfully saved.



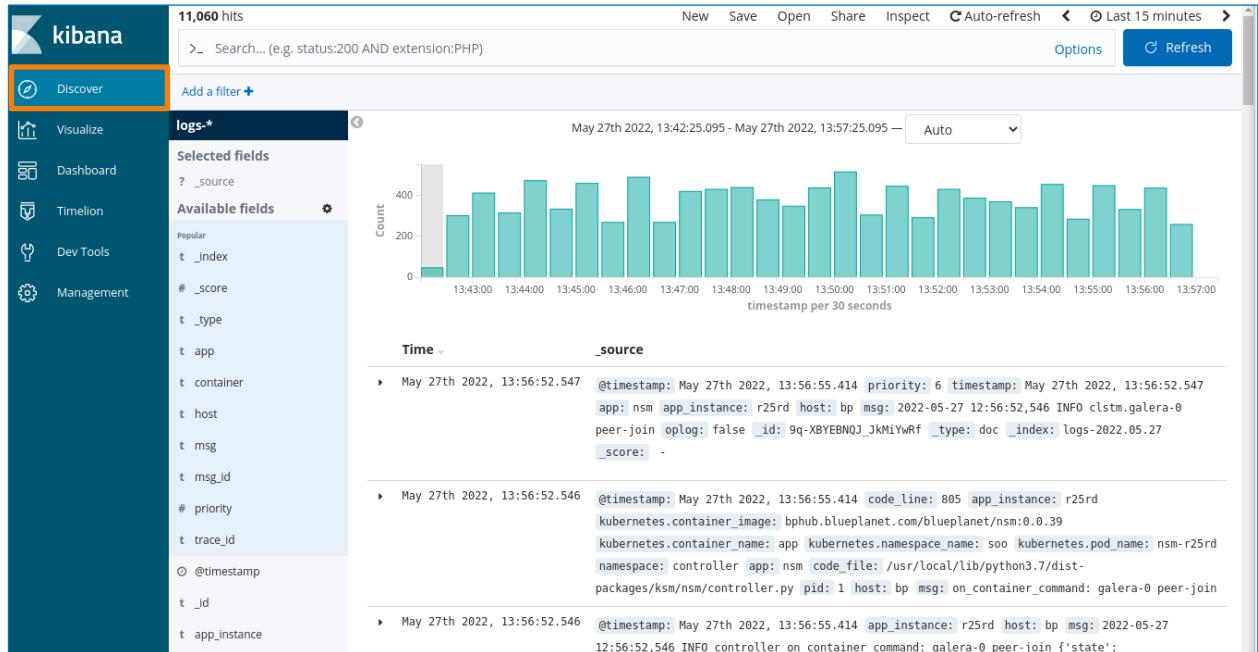
Task 6: Exploring Blue Planet Logs with Kibana

In this task, you will use Kibana to explore the Blue Planet platform logs as well as create new visualizations and dashboards in Kibana.

1. From your browser Blue Planet UI session, select **System > Platform > Logging** to access Kibana.

The screenshot shows the Blue Planet UI interface. On the left, there is a vertical navigation bar with various icons and links. The 'Platform' section is expanded, and 'Logging' is selected, highlighted with an orange box. Another orange box highlights the 'Logs' icon in the navigation bar. The main content area is titled 'kibana' and contains sections for 'Discover', 'Visualize', 'Dashboard', 'Timelion', 'Dev Tools', and 'Management'. A 'Recently viewed' section shows 'Messages per App' and 'Messages per day'. Below this is a 'Add Data to Kibana' section with two cards: 'APM' and 'Logging'. The 'APM' card has a description and an 'Add APM' button. The 'Logging' card has a description and an 'Add log data' button. The overall theme is blue and white, consistent with the Ciena branding.

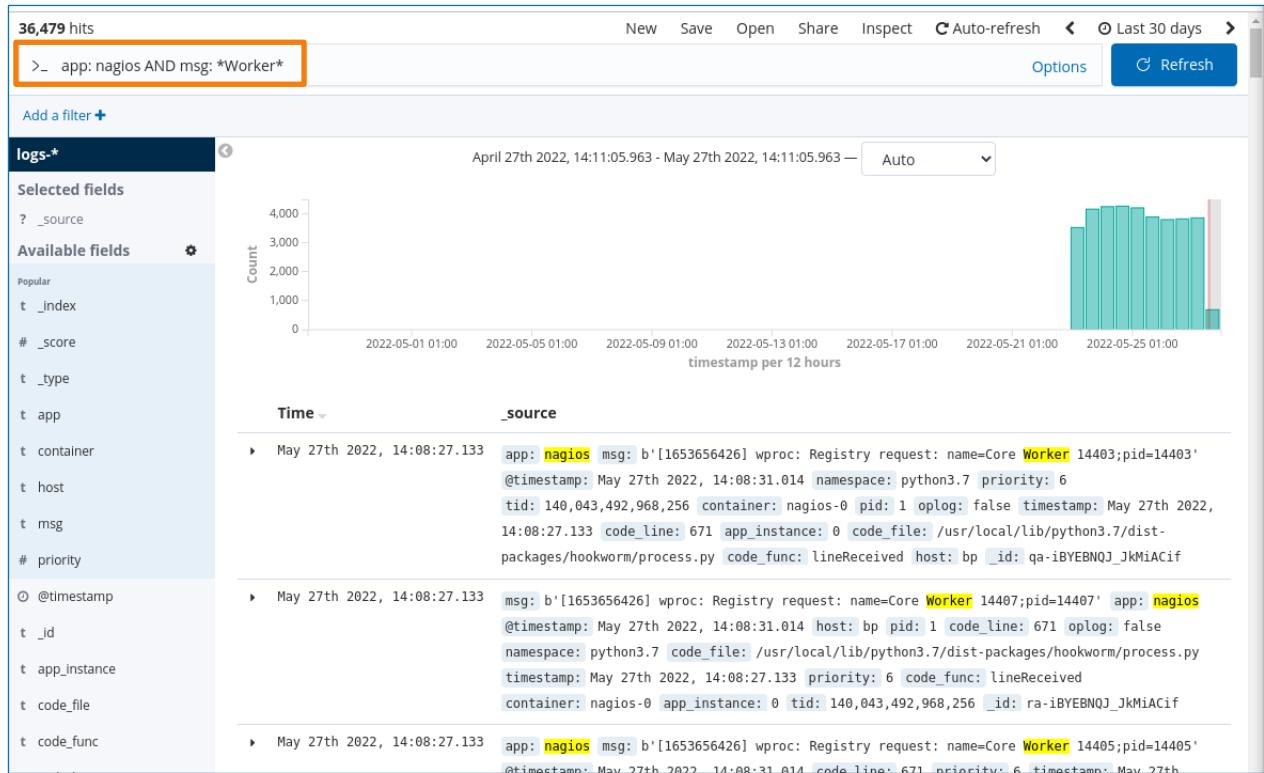
2. From the Kibana menu, choose **Discover**. The Discover page lists all logs that are captured from the system in chronological order. Kibana features powerful searching capabilities so you can locate messages of interest.



3. Before searching for log messages, set the time frame to **Last 30 days**.

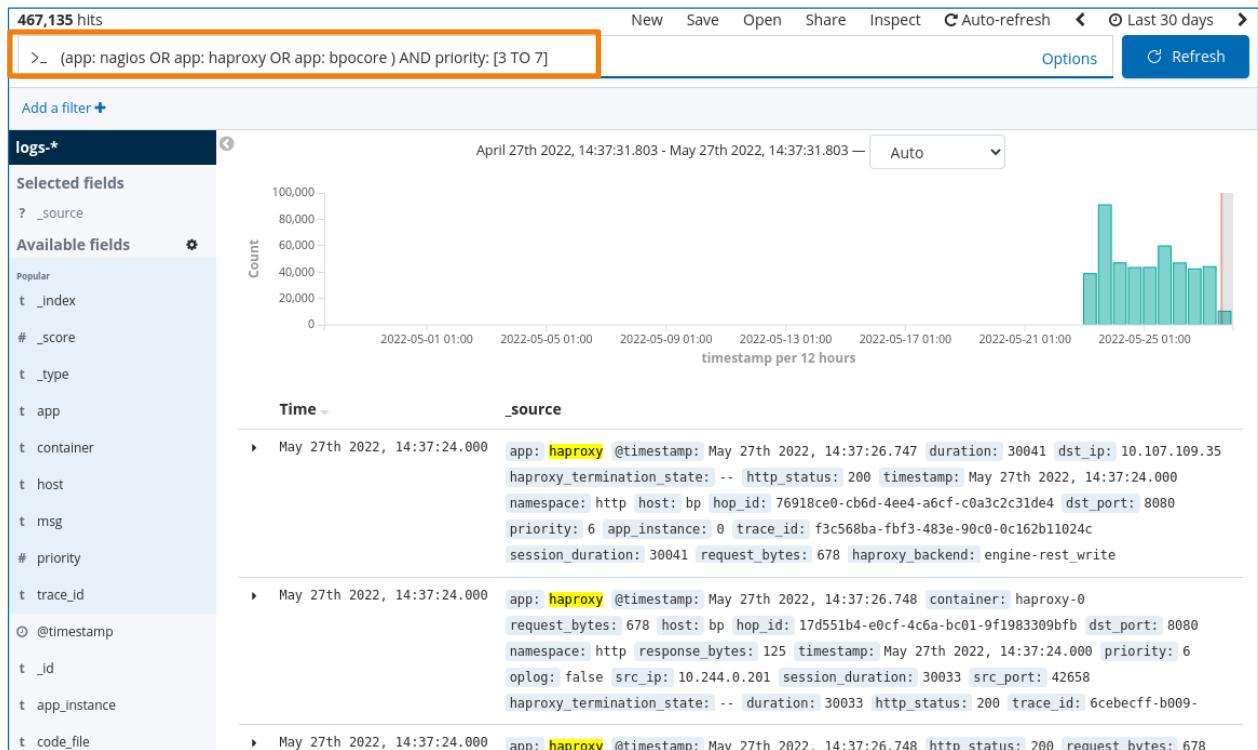


4. In the **Search** bar, enter the following text: **app: nagios AND msg: *Worker*** and then click the **Update** button. Note how you can use logical operators (AND) as well as wildcards for the text. The **msg** field contains the original log message, so in this scenario, we are searching for messages originated by the Nagios app, but that also have the text “Worker” anywhere in the message.

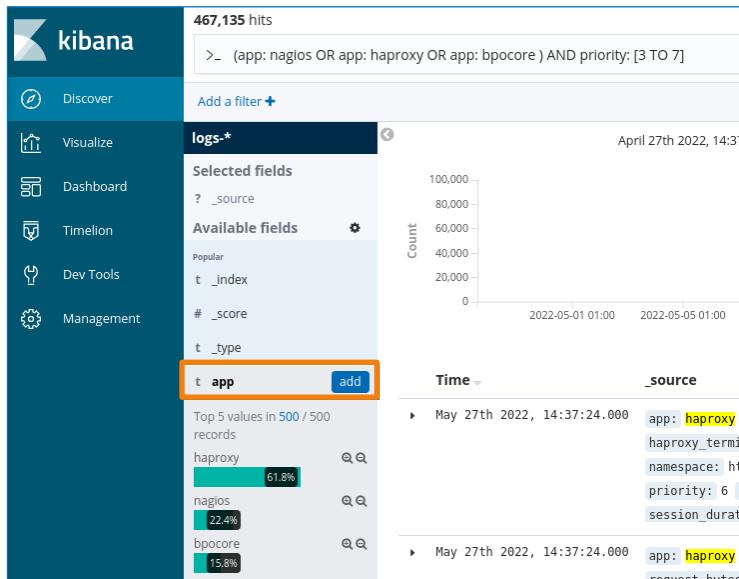


NOTE: The output from your Kibana might vary due to different timestamps and availability and the number of specific log messages recorded in the past.

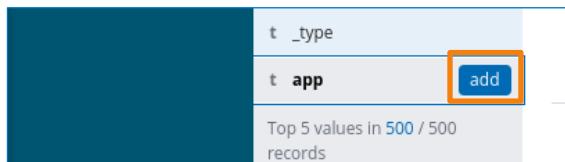
- Try a more complex search. Display all messages coming from the apps Nagios, haproxy, or bpocore that have the priority number in the range of 3 to 7. Modify the search text to the following string: **(app: nagios OR app: haproxy OR app: bpocore) AND priority: [3 TO 7]** and click **Update**.



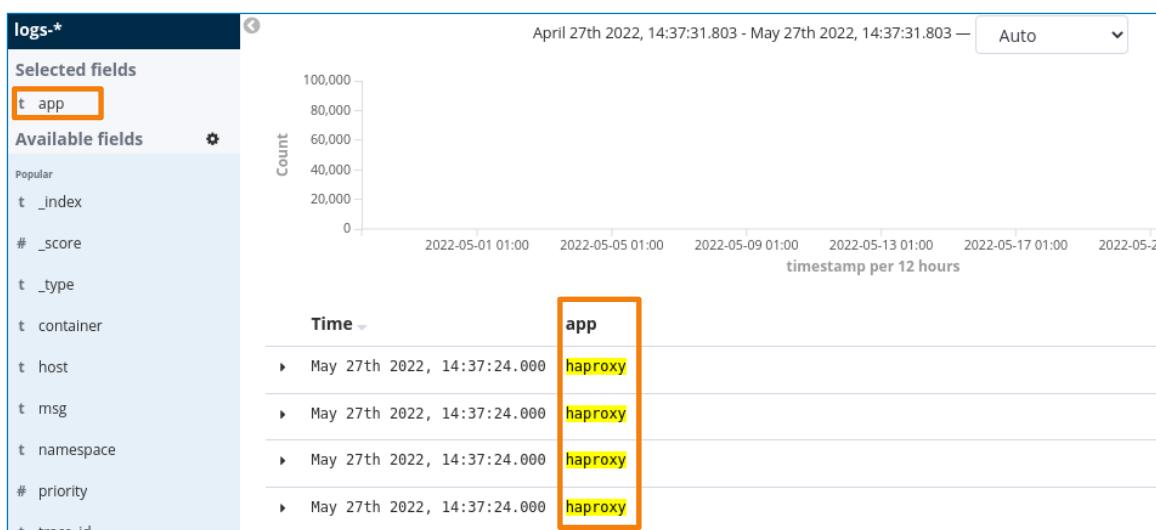
6. Note that on the left side of the log messages, there is a column named **logs-***. By default, the chosen fields show only **_source**, which means to show the source of the log messages, with all data. You can customize the displaying of the data by choosing fields in the **Available fields** section. In **Available fields**, click **app**.



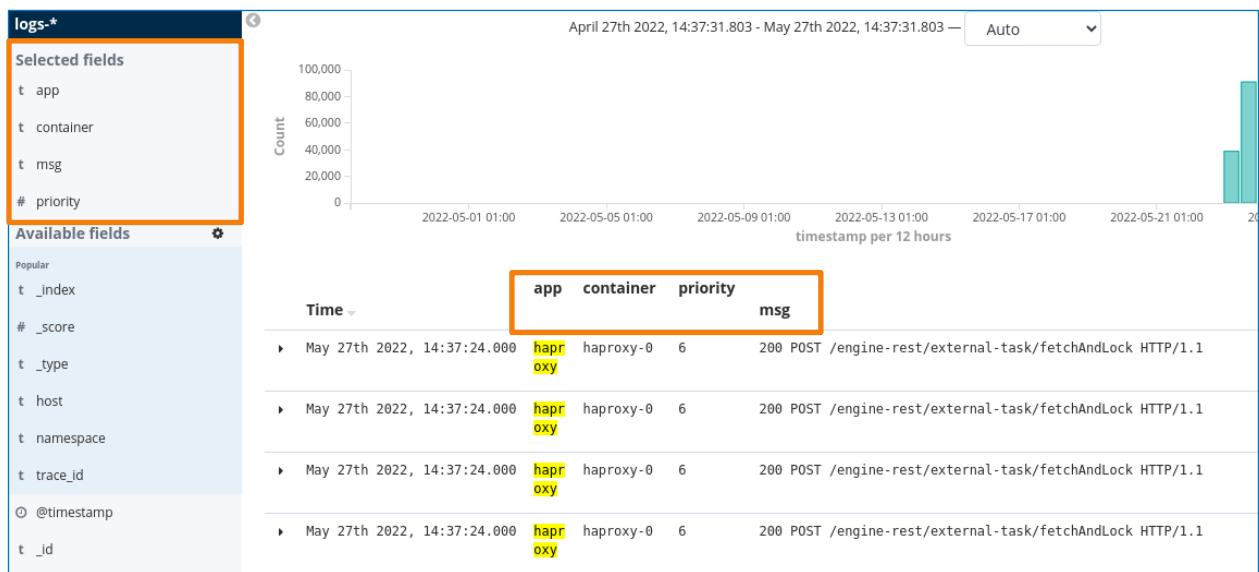
7. Notice when you click the **app** field, the section expands, giving you the statistical distribution of the messages coming from these three apps. 61,8% of messages are from haproxy, 22,4% from Nagios, and 15,8% from bpocore. Your numbers may vary. Now click the **add** button next to the **app** field.



8. Note how the **app** field moved to the **Selected fields** section and the log messages only display the app column.



9. Customize the log message display area to include these fields: **container**, **priority**, and **msg**, by pressing the **add** button next to the said fields in the **Available fields** section.



NOTE: Observe that your log message list now shows the four selected values only.

- You can additionally filter the results within your initial search parameters. Expand the first log message to show field details.

Time	msg
May 27th 2022, 14:37:24.000 <input checked="" type="checkbox"/> haproxy	haproxy-0 6 200 POST /engine-rest/external-task/fetchAndLock HTTP/1.1

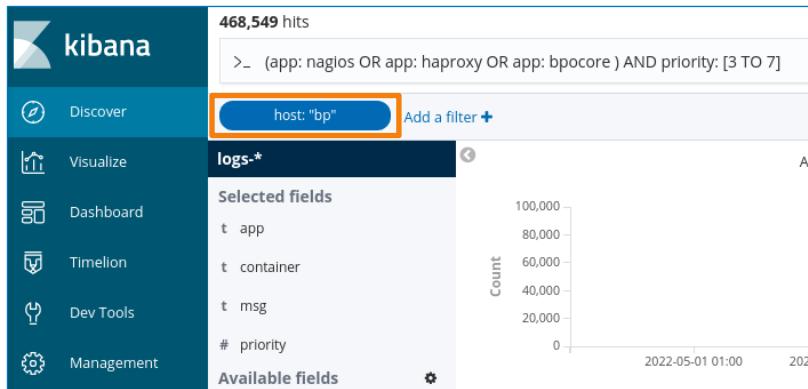
Table [JSON](#) [View surrounding documents](#) [View single document](#)

Field	Type	Value
@timestamp	date	May 27th 2022, 14:37:26.747
_id	string	0a-8BYEBNQJ_JkMifJDC
_index	string	logs-2022.05.27
_score	float	-
_type	string	doc
app	string	haproxy
app_instance	string	0
container	string	haproxy-0
content_type	string	application/json
dst_connect_time	float	0
dst_ip	string	10.107.109.35
dst_port	float	8080
dst_response_time	float	30040
duration	float	30041
haproxy_backend	string	engine-rest_write
haproxy_termination_state	string	--
hop_id	string	76918ce0-cb6d-4ee4-a6cf-c0a3c2c31de4
host	string	bp <input checked="" type="checkbox"/>
http_status	float	200
msg	string	200 POST /engine-rest/external-task/fetchAndLock HTTP/1.1
namespace	string	http

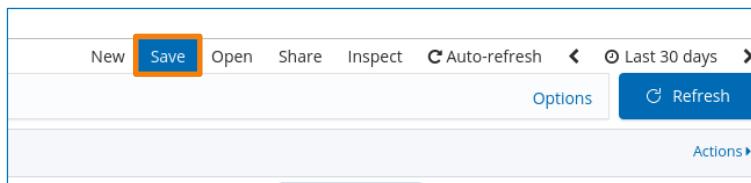
- Click the + button next to the host to add this field and its value to the filter.

t hop_id	76918ce0-cb6d-4ee4-a6cf-c0a3c2c31de4
t host	bp <input checked="" type="checkbox"/>
? http_status	200

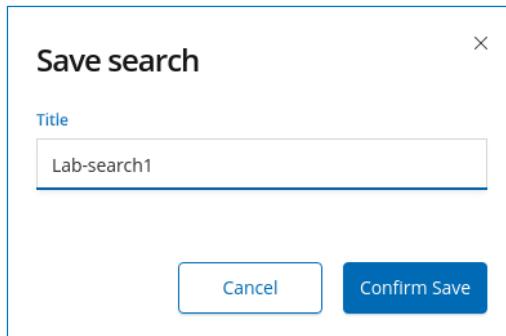
12. The filter **host: "bp"** is now added to the filter list. The resulting messages must conform both to the search condition as well as the filters.



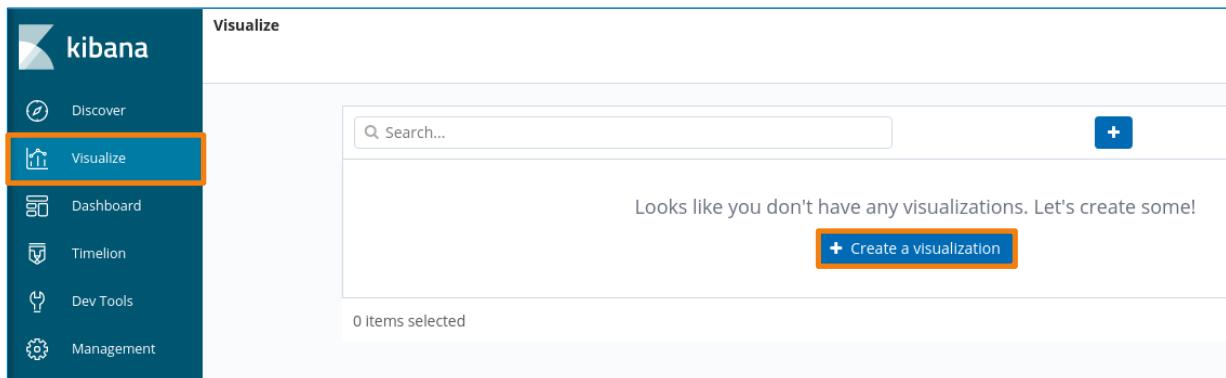
13. Click **Save** from the top menu to save your search.



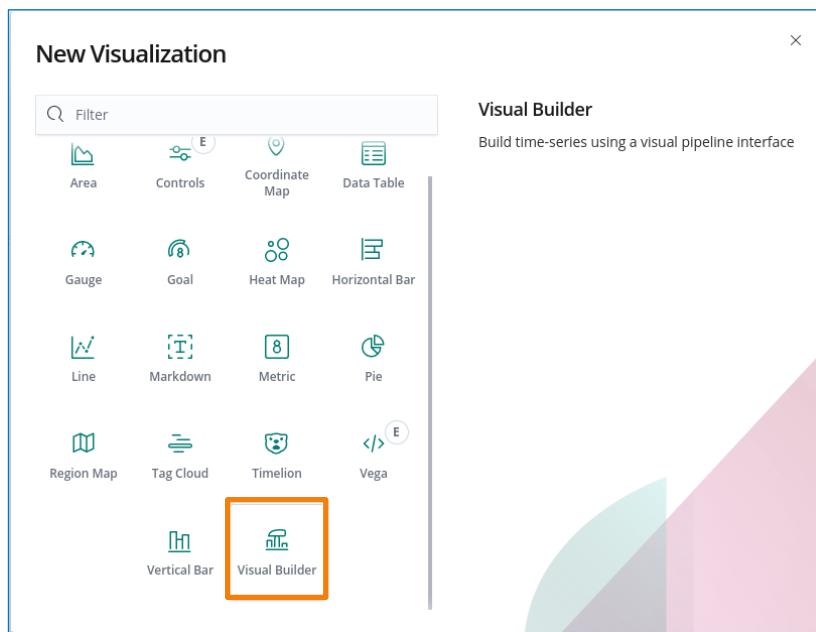
14. In the popup window, enter **Lab-search1** and press **Confirm Save**.



15. In Kibana, you can visualize the data from the Blue Planet Platform logs. Click **Visualize** from the main Kibana menu, then click the **Create a visualization** button.



16. From the **New Visualization** window, select **Visual Builder**.



17. A new visualization window will open. Configure a graph that will show the top 10 applications by log messages volume. First set the time frame to **Last 7 days**. Next, in the **Data** pane, set the following parameters:

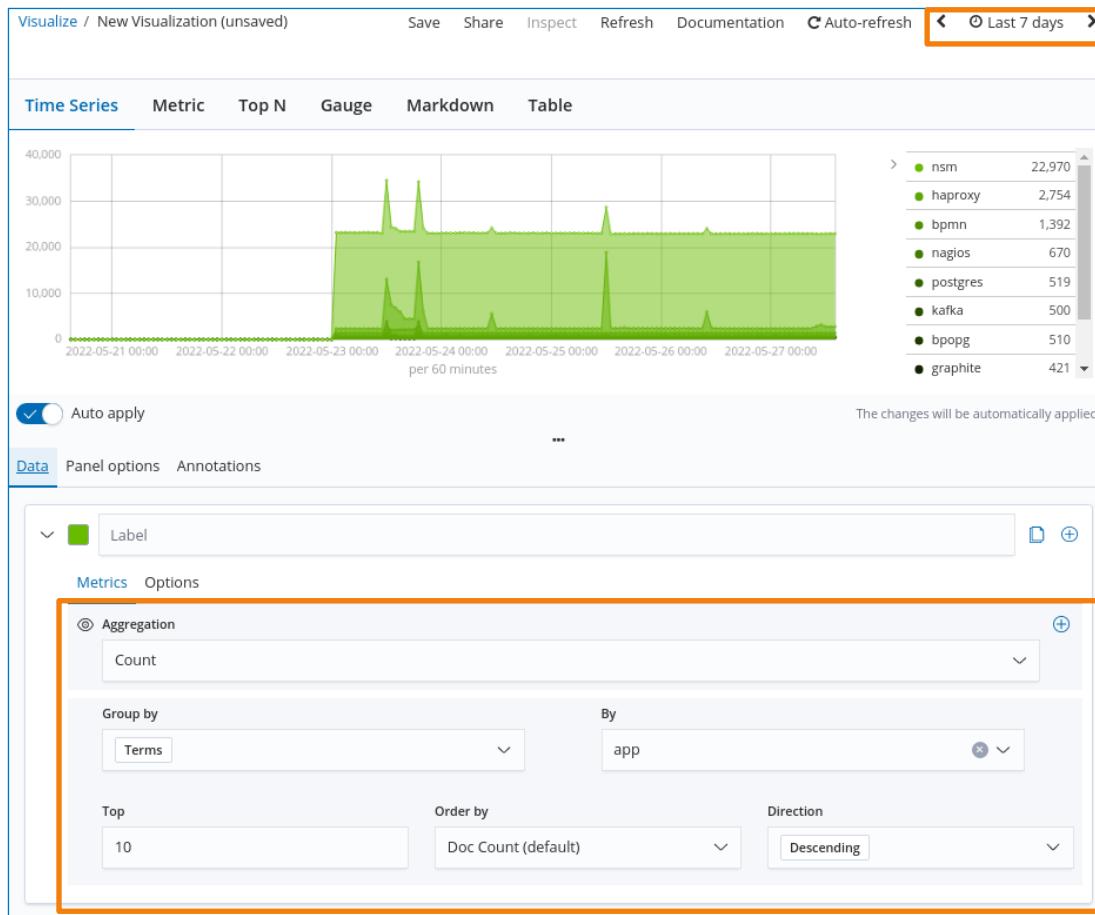
Aggregation: **Count**

Group by: **Terms**

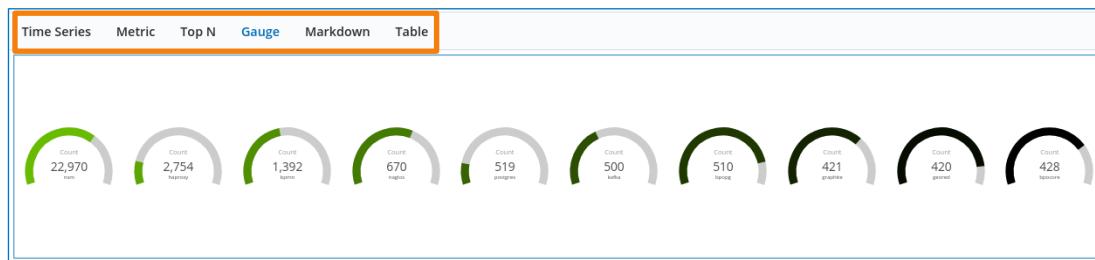
By: **app**

Top: **10**

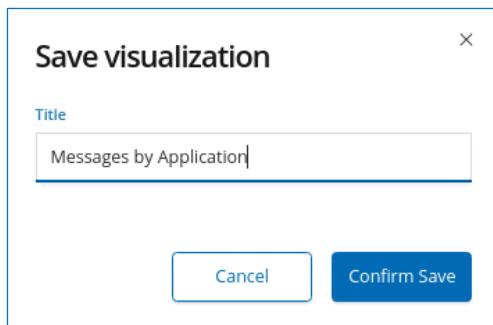
Leave other parameters as they are.



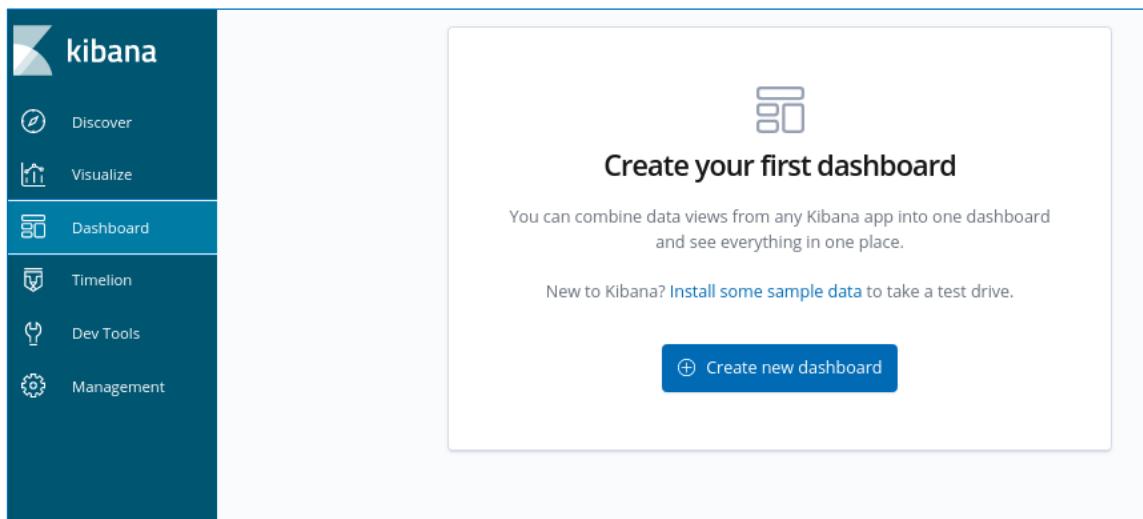
- To try out a different graphical view of the results, click the **Metric**, **Top N**, **Gauge**, and other options from the top menu. When done testing, click the **Time Series** before you continue to the next step.



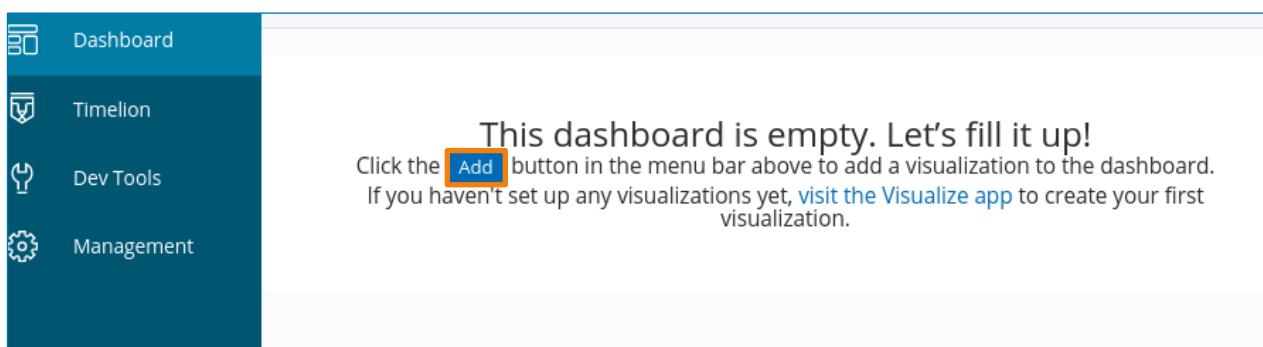
19. From the top menu, click **Save** to save the newly created visualization. When asked for the name, enter **Messages by Application** and press **Confirm Save**.



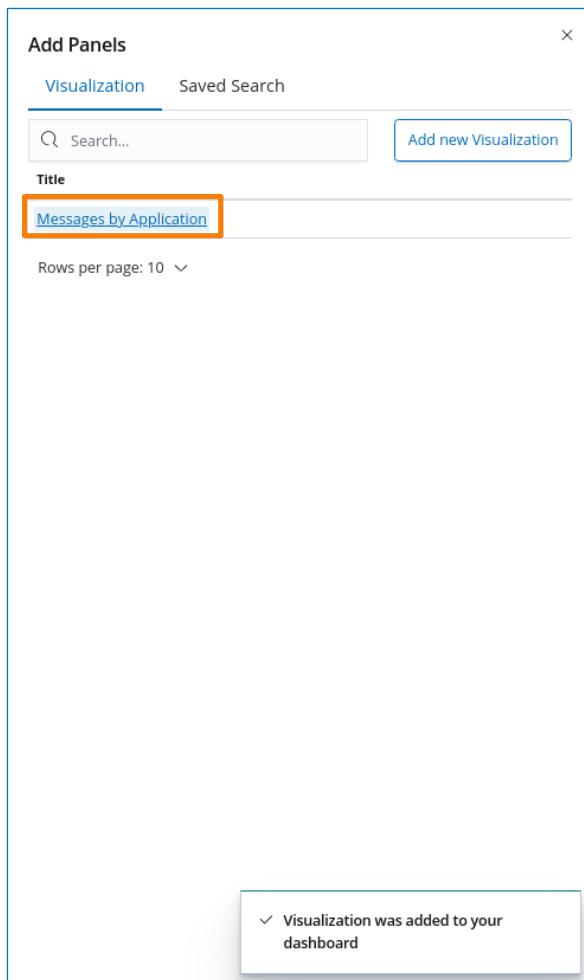
20. From the main Kibana menu, choose the **Dashboard** option and then click the **Create new dashboard** button to create your custom dashboard.



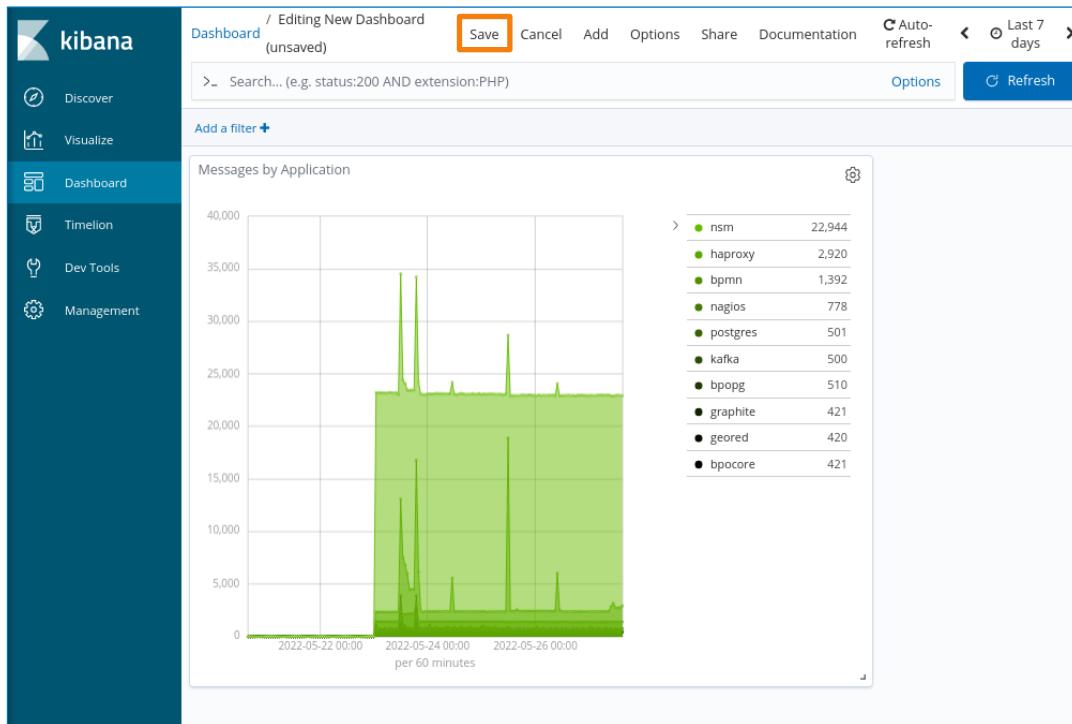
21. Next, click the **Add** button.



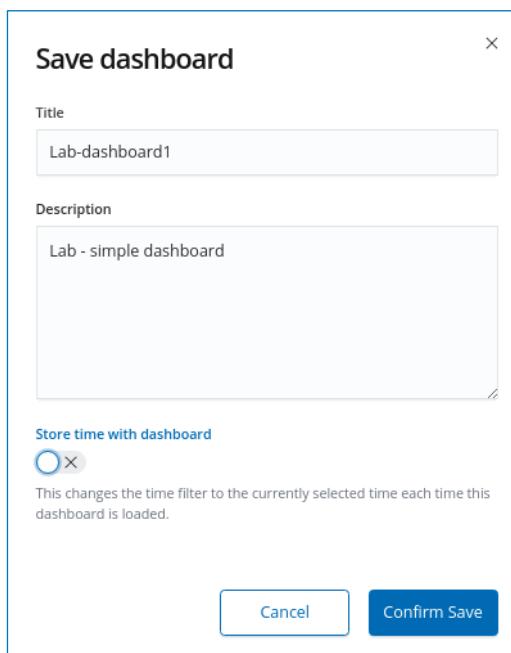
22. The **Add Panels** window opens. Click the **Messages by Application** to add your visualization to the dashboard.



23. Press the **X** button in the top right corner to close the **Add Panels** window. Your dashboard now has a new panel with the visualization that you created previously. Now save the dashboard by clicking **Save** in the top menu.



24. Enter the following parameters before you save the dashboard, then press **Confirm Save**:
- Title:** Lab-dashboard1
- Description:** Lab – simple dashboard



Save dashboard

Title
Lab-dashboard1

Description
Lab - simple dashboard

Store time with dashboard This changes the time filter to the currently selected time each time this dashboard is loaded.

Cancel Confirm Save

End of Lab