IBM Cloud

Cloud Integration

Security and Compliance Management with IBM Cloud Pak for Multicloud Management

Contents

Introduction

To meet the unique needs of your business and remain competitive in today's fast-moving environment, you may find yourself adopting infrastructure and solutions from a wide range of cloud vendors.

A hybrid, multicloud world is quickly becoming the new normal. But managing your cloud-based services and data across multiple providers can feel overwhelming. With each set of cloud services coming with its own tools, you're likely facing increased complexity and cost. New management solutions and delivery methods can help optimize performance, control costs, provide quick cloud access and secure your mix of applications, environments, and data, whether they are inside your data center or in the cloud.

IBM Cloud Pak for Multicloud Management can manage Kubernetes clusters that are deployed on any target infrastructure - either in your own data center or in a public cloud. IBM Cloud Pak for Multicloud Management includes IBM Cloud App Management to simplify monitoring your applications across any cloud environment.

IBM Cloud Pak for Multicloud Management helps companies make the transition from traditional monitoring systems to cloud-based ones more easily. It effectively monitors all kinds of IT resources in a hybrid environment. It helps Operation teams manage hybrid environments without hiring new personnel to support each new technology that is being used by developers.

Cloud Pak for Multicloud Management provides consistent visibility, automation, and governance across a range of multicloud management capabilities such as cost and asset management, infrastructure management, application management, multi-cluster management, edge management, and integration with existing tools and processes. Customers can leverage Cloud Pak for Multicloud Management to simplify their IT and application ops management, while increasing flexibility and cost savings with intelligent data analysis driven by predictive signals.

This Tutorial explores how to use governance and compliance features to manage your multicloud environments with a consistent set of configuration and security policies across all applications and clusters.You explore the following key capabilities:

- Understand Cloud Pak Policy and Governance
- Learn to create and customize policies with the out of the box policy templates
- Learn to use namespace policies
- Learn to use network policies

For more information about IBM Cloud Pak for Multicloud Management, visit:
https://www.ibm.com/cloud/cloud-pak-for-management

Business Scenario

As a member of the Security Operation (SecOps) team, you are having problems to minimize risks and identify policies violations in your multicloud hybrid world. Manage a Security Policy for all your cloud-based services and data across multiple providers is overwhelming your team.
Your company is deploying multiple Kubernetes clusters to address their specific needs. Some Dev teams are deploying clusters across public and private clouds, and some are deploying clusters across regions, and some are deploying clusters to support the development and test needs.

As different teams deploy more clusters, new challenges are introduced:

- How do I set consistent security policies across environments?
- Which clusters are compliant?

Because of that, you want to explore how IBM Cloud Pak for Multicloud Management, provides consistent visibility, governance and automation of your complex environment.

IBM Cloud Pak for Multicloud Management Governance and risk dashboard allows you to view and manage the number of security risks and policy violations in your clusters and applications. Policy templates are used to create one or more policies for third party or external security controls. For example, you can create a mutation policy with the mutation policy controller. Each policy document can have at least one or multiple templates.

By using policy-based role and compliance management, you are able to:

- Set and enforce polices for security, applications, and infrastructure or auto enforcement at the cluster level.
- Check compliance against deployment parameters, configuration, and policies.
- Automatically remediate violations.

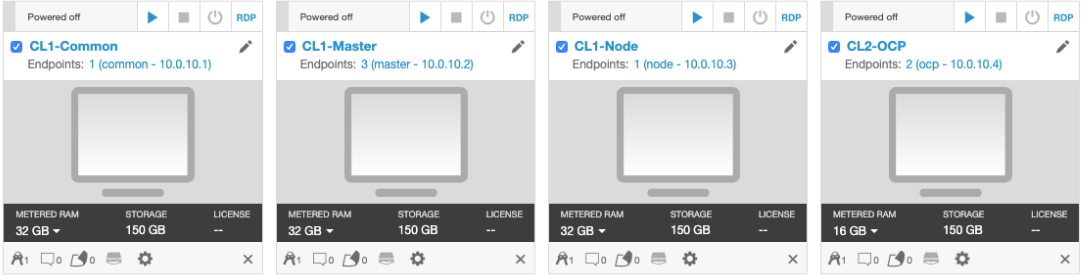
In this tutorial, you create and enforce the following governance policies:

- Namespace policy
- Network policy

Tutorial Environment

Environment Overview

Four virtual machines have been provided for this tutorial.



Powered off RDP
☑ CL1-Common
Endpoints: 1 (common - 10.0.10.1)
METERED RAM 32 GB · STORAGE 150 GB · LICENSE --

Powered off RDP
☑ CL1-Master
Endpoints: 3 (master - 10.0.10.2)
METERED RAM 32 GB · STORAGE 150 GB · LICENSE --

Powered off RDP
☑ CL1-Node
Endpoints: 1 (node - 10.0.10.3)
METERED RAM 32 GB · STORAGE 150 GB · LICENSE --

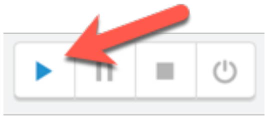
Powered off RDP
☑ CL2-OCP
Endpoints: 2 (ocp - 10.0.10.4)
METERED RAM 16 GB · STORAGE 150 GB · LICENSE --

- CL1 is the Hub Cluster. It is an OpenShift Container Platform, **OCP.** It is installed in three VMs, the **CL1-Master** VM with master node, **CL1-Common** VM with all common services, and one compute node VM: **CL1-Node**. As part of this tutorial, you use CL1-Master as workstation too.
- CL2 is the Managed Cluster. It is an All-in-One OpenShift cluster. Here you have **CL2-OCP** VM.

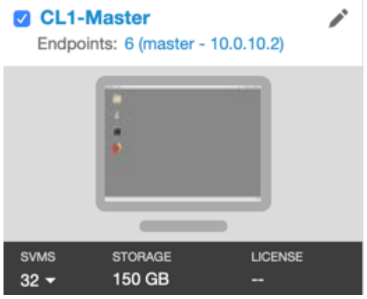
Starting your Environment

__1. Follow the instructions on the right to provision the lab environment.

__2. If the environment is not started automatically, click the play button in the upper right to start all four virtual machines. This takes approximately 15 minutes.

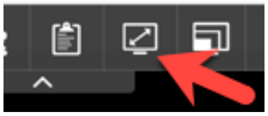


__3. Click **CL1-Master** to access the desktop of the server



☑ CL1-Master
Endpoints: 6 (master - 10.0.10.2)
SVMS 32 · STORAGE 150 GB · LICENSE --

__4. A Linux desktop appears in your browser tab. Feel free to resize the window for a better view.

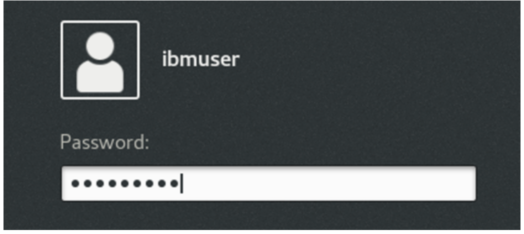


Lab - IBM Cloud Pak for Multicloud Management Cluster Management

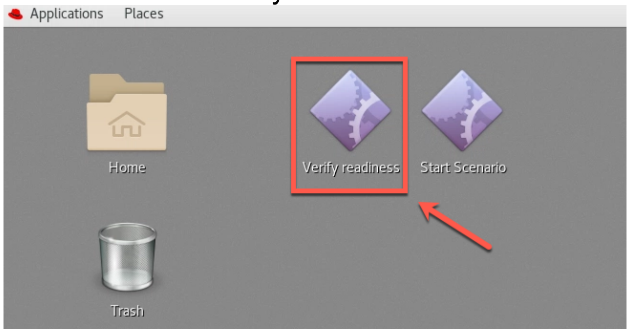
4.1 Accessing the IBM Cloud Pak for Multicloud Management

In this section, you explore your IBM Cloud Pak for Multicloud Management environment.

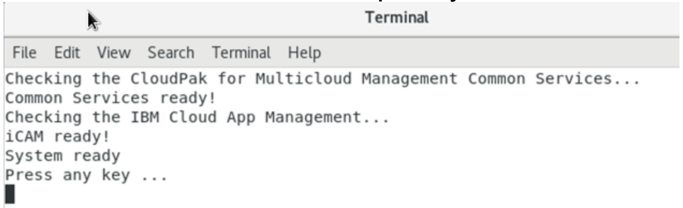
__1. Log in as **ibmuser** using **passw0rd** as a password.



__2. Verify that the environment was fully initialized. On the desktop, you should see an icon named **Verify readiness**. Double-click on the icon to run the verification script that checks if the environment was fully initialized.



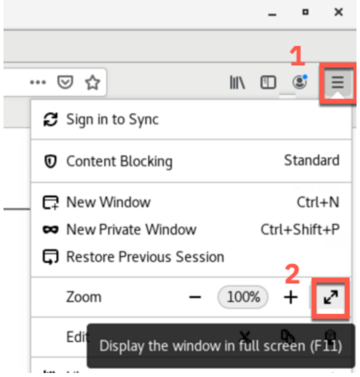
__3. In the terminal window that opens you should see the following text:



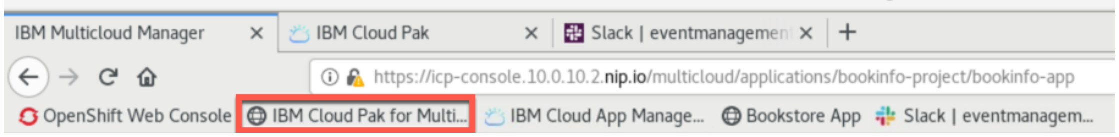
```
Checking the CloudPak for Multicloud Management Common Services...
Common Services ready!
Checking the IBM Cloud App Management...
iCAM ready!
System ready
Press any key ...
```

If the environment is not ready, wait until the "System ready" message is displayed.

__4. Start the **Firefox** browser (link is on the desktop). For better viewing, switch the browser in the virtual machine into the "Fullscreen" mode.

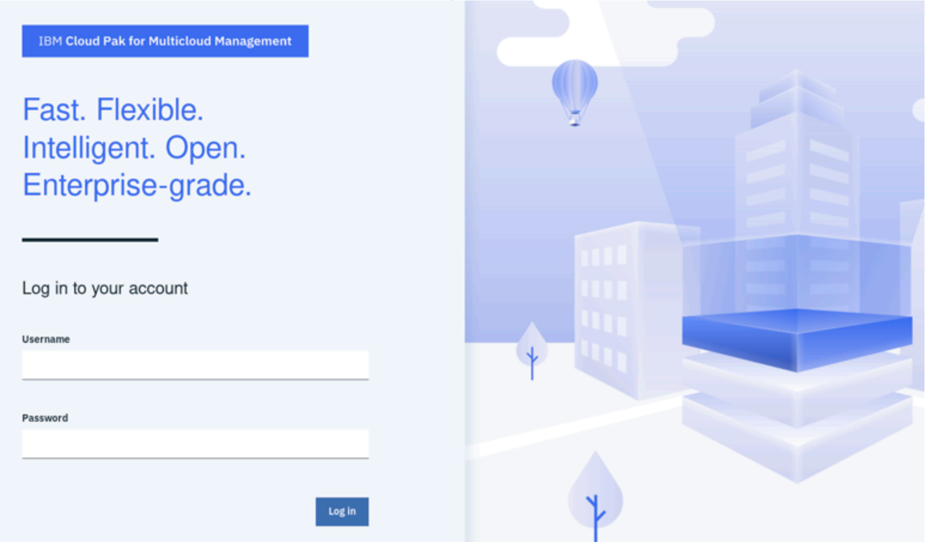


__5. The link to the IBM Cloud Pak for Multicloud Management is added to the Bookmark toolbar. Click the bookmark to open the UI.



HINT: At any point during the lab, if you are lost navigating in the Cloud Pak UI you can click the link again to return to the main product screen.

__6. If not already logged in, log in as **admin** with a password of **Passw0rd!**
Maximize the window, if not already maximized.



IBM Cloud Pak for Multicloud Management

Fast. Flexible.
Intelligent. Open.
Enterprise-grade.

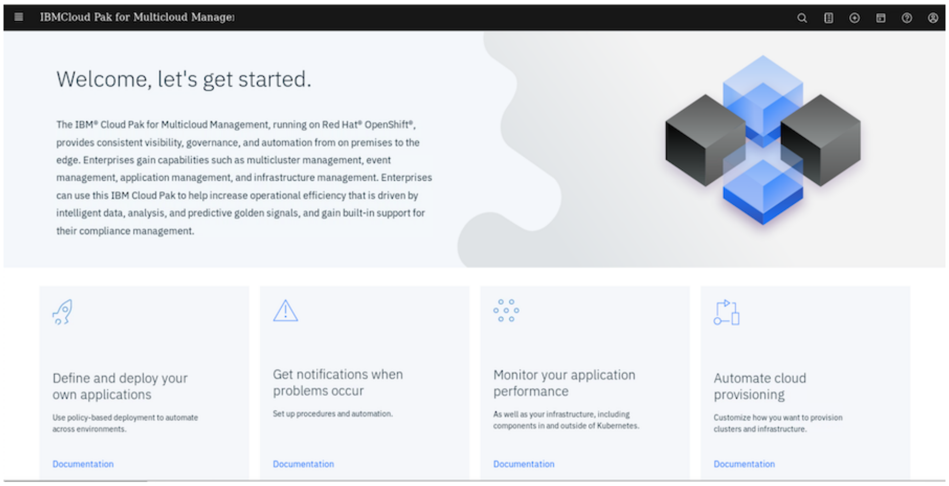
Log in to your account

Username

Password

Log in

__7. After you log in, you are presented with a Welcome screen. If you see a different screen (Authentication), click again on the IBM Cloud Pak for Multicloud Management link on the bookmark toolbar.



IBMCloud Pak for Multicloud Manage

Welcome, let's get started.

The IBM® Cloud Pak for Multicloud Management, running on Red Hat® OpenShift®, provides consistent visibility, governance, and automation from on premises to the edge. Enterprises gain capabilities such as multicluster management, event management, application management, and infrastructure management. Enterprises can use this IBM Cloud Pak to help increase operational efficiency that is driven by intelligent data, analysis, and predictive golden signals, and gain built-in support for their compliance management.

Define and deploy your own applications

Use policy-based deployment to automate across environments.

Documentation

Get notifications when problems occur

Set up procedures and automation.

Documentation

Monitor your application performance

As well as your infrastructure, including components in and outside of Kubernetes.

Documentation

Automate cloud provisioning

Customize how you want to provision clusters and infrastructure.

Documentation

4.2 Create a namespace policy

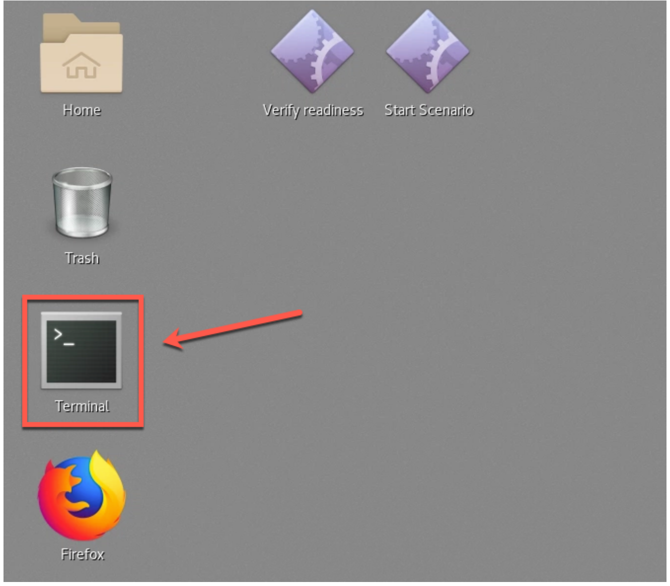
Kubernetes namespaces help organize cluster resources between multiple users and split the resource quote. Cluster administrator might restrict the user to use specific namespaces for applications. The namespace policy allows you to catch cluster violations when namespaces are not defined as per the policies.

A sample namespace policy resemble the following:

```
apiVersion: policy.mcm.ibm.com/v1alpha1
kind: Policy
metadata:
  name: policy-namespace-1
  namespace: mcm
spec:
  complianceType: musthave
  remediationAction: inform
  namespaces:
    exclude: ["kube-*"]
    include: ["default"]
  object-templates:
    - complianceType: musthave
      objectDefinition:
        kind: Namespace # must have namespace 'prod'
        apiVersion: v1
        metadata:
          name: prod
     ...
```

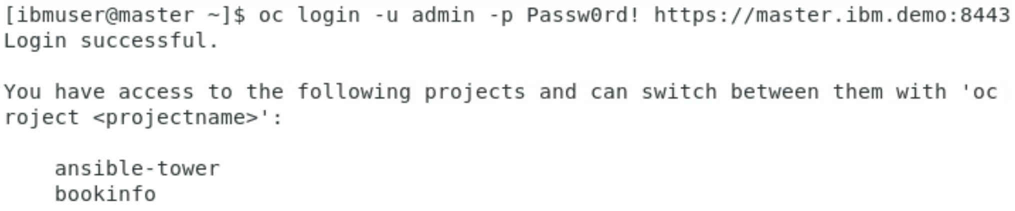
In this section, you create a policy that ensures that a specified namespace is present in clusters that match the selection criteria.

__1. Open Terminal window.



__2. Log in to OpenShift cluster using the command below:

oc login -u admin -p Passw0rd! https://master.ibm.demo:8443



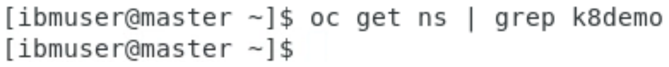
```
[ibmuser@master ~]$ oc login -u admin -p Passw0rd! https://master.ibm.demo:8443
Login successful.

You have access to the following projects and can switch between them with 'oc
roject <projectname>':

    ansible-tower
    bookinfo
```

__3. Next steps, you create a Policy to inform/enforce a namespace in your cluster. Let's verify that you don't have this namespace by now. Run the command below:

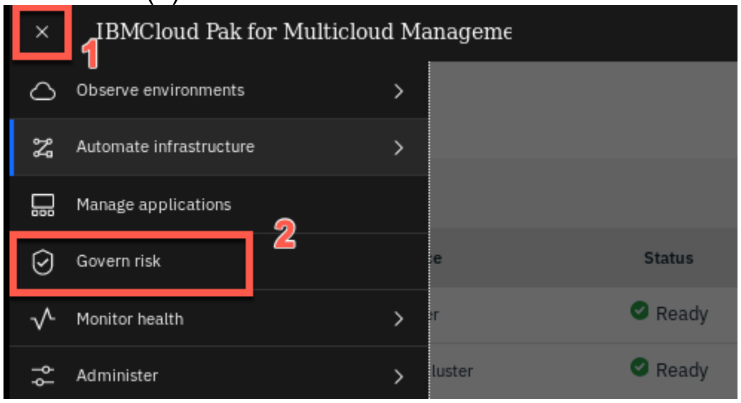
oc get ns | grep k8demo



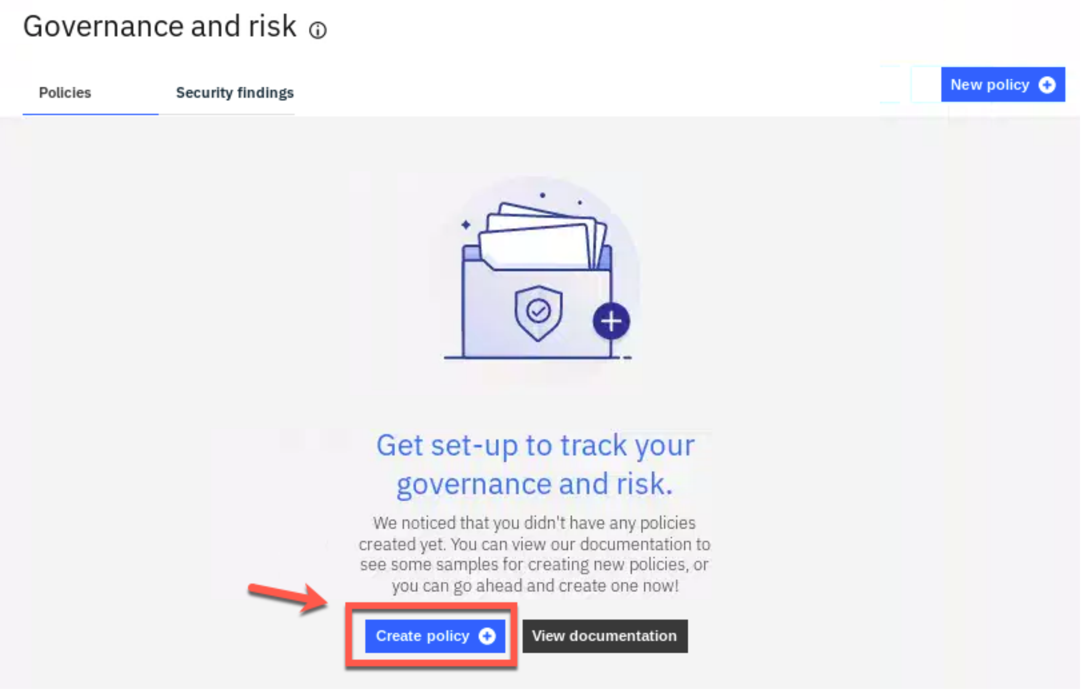
```
[ibmuser@master ~]$ oc get ns | grep k8demo
[ibmuser@master ~]$
```

Great! So far, you don't have the k8demo namespace. Let's create a Policy to inform when your cluster is not compliance with a namespace policy.

__4. Back to the MCM web page on Firefox. On the top-left of the page, open the **Menu** (1) and select **Govern risk** (2).



__5. Here you see the Policy tab. This view displays the policies that have been created and the dashboard of policy compliance for each cluster. By now, you don't have any Policy created. Let's do it! Click **Create policy**.

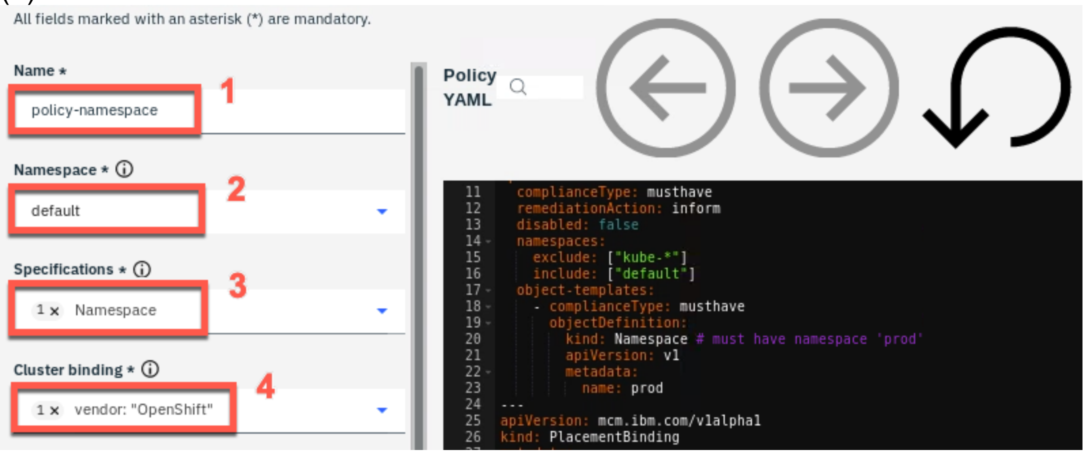


Governance and risk ⓘ

Get set-up to track your governance and risk.

We noticed that you didn't have any policies created yet. You can view our documentation to see some samples for creating new policies, or you can go ahead and create one now!

__6. On the Name field type **policy-namespace** (1), on the Namespace field select **default** (2)**,** on the Specifications field select **namespace** (3) and on Cluster binding select **vendor: OpenShift** (4).



All fields marked with an asterisk (*) are mandatory.

Name * policy-namespace

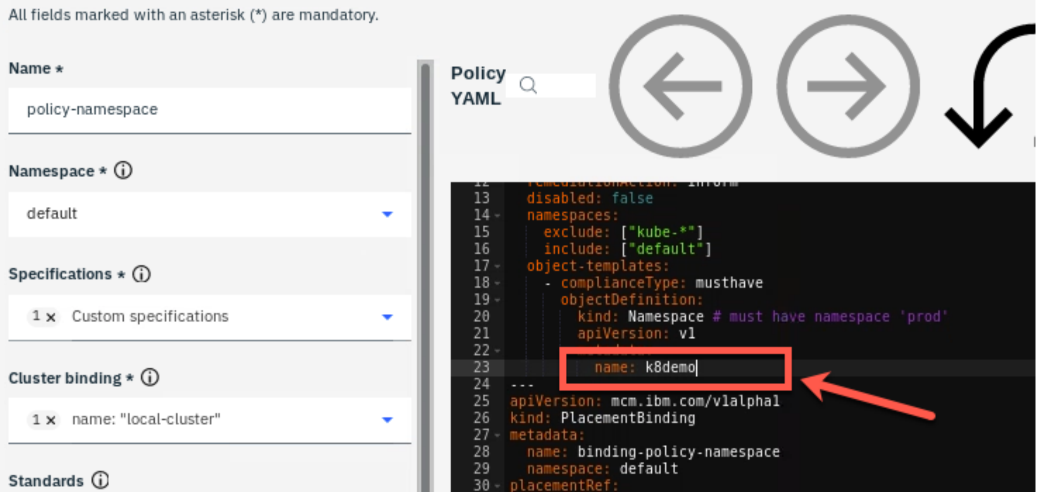
Namespace * ⓘ default

Specifications * ⓘ 1 × Namespace

Cluster binding * ⓘ 1 × vendor: "OpenShift"

```
  complianceType: musthave
  remediationAction: inform
  disabled: false
  namespaces:
    exclude: ["kube-*"]
    include: ["default"]
  object-templates:
    - complianceType: musthave
      objectDefinition:
        kind: Namespace # must have namespace 'prod'
        apiVersion: v1
        metadata:
          name: prod
---
apiVersion: mcm.ibm.com/v1alpha1
kind: PlacementBinding
```

__7. Now, let's change the namespace name value. In the yaml file section, on the right, change the name attribute from **prod** to **k8demo**. With that, you are creating a Policy to verify if you have a k8demo namespace/project in your cluster.



All fields marked with an asterisk (*) are mandatory.

Name * policy-namespace

Namespace * default

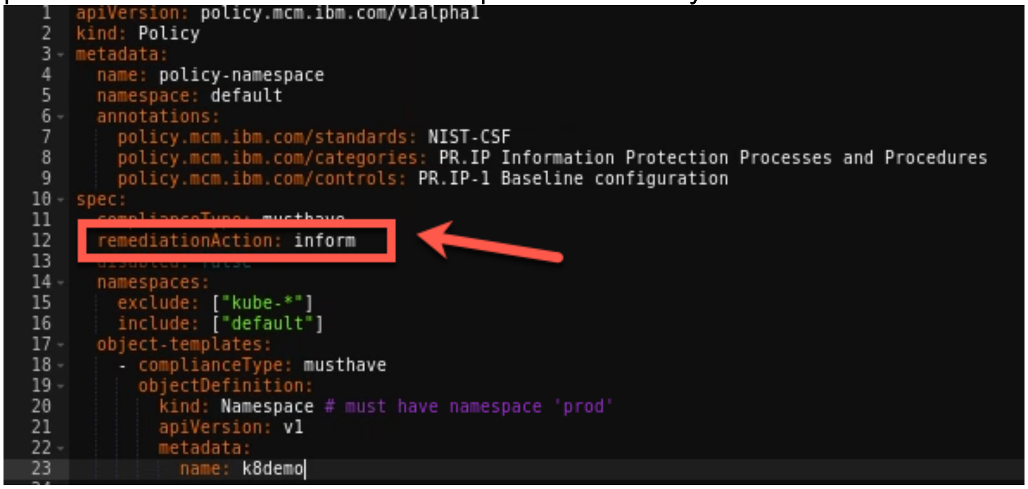
Specifications * 1 × Custom specifications

Cluster binding * 1 × name: "local-cluster"

Standards

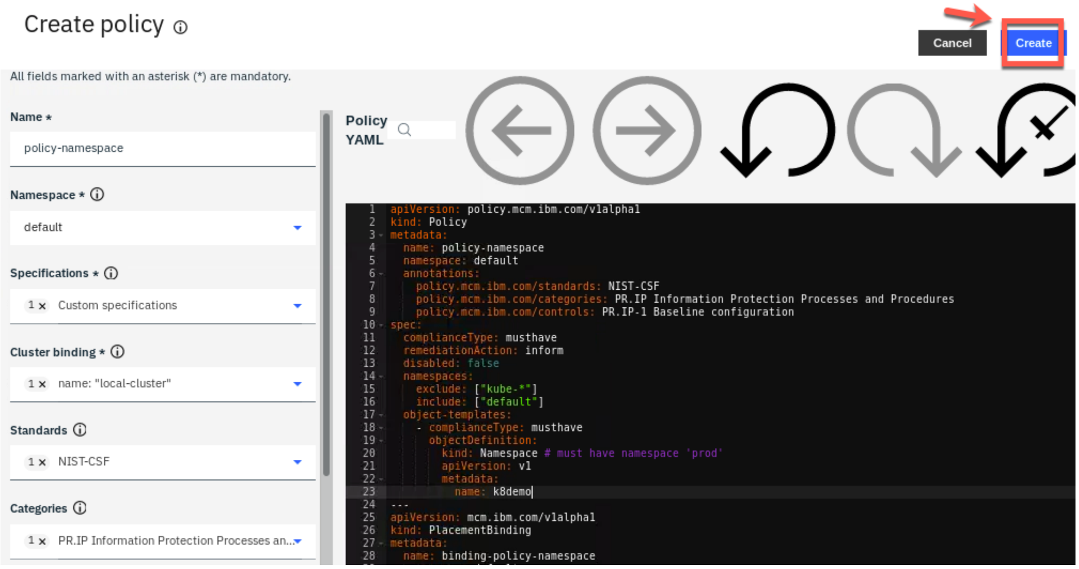
```
    remediationAction: inform
    disabled: false
    namespaces:
      exclude: ["kube-*"]
      include: ["default"]
    object-templates:
      - complianceType: musthave
        objectDefinition:
          kind: Namespace # must have namespace 'prod'
          apiVersion: v1
          metadata:
            name: k8demo
---
apiVersion: mcm.ibm.com/v1alpha1
kind: PlacementBinding
metadata:
  name: binding-policy-namespace
  namespace: default
placementRef:
```

__8. Notice that the policy is set to **inform** rather than enforce. With value inform, the policy only reports whether the cluster is compliant to the specified policies. With value enforce, the policy provides automatic remediation. Keep inform value by now.



```
apiVersion: policy.mcm.ibm.com/v1alpha1
kind: Policy
metadata:
  name: policy-namespace
  namespace: default
  annotations:
    policy.mcm.ibm.com/standards: NIST-CSF
    policy.mcm.ibm.com/categories: PR.IP Information Protection Processes and Procedures
    policy.mcm.ibm.com/controls: PR.IP-1 Baseline configuration
spec:
  complianceType: musthave
  remediationAction: inform
  disabled: false
  namespaces:
    exclude: ["kube-*"]
    include: ["default"]
  object-templates:
    - complianceType: musthave
      objectDefinition:
        kind: Namespace # must have namespace 'prod'
        apiVersion: v1
        metadata:
          name: k8demo
```

__9. Click the button **Create** to create your new policy.



Create policy ⓘ

All fields marked with an asterisk (*) are mandatory.

Name *
policy-namespace

Namespace * ⓘ
default

Specifications * ⓘ
1 × Custom specifications

Cluster binding * ⓘ
1 × name: "local-cluster"

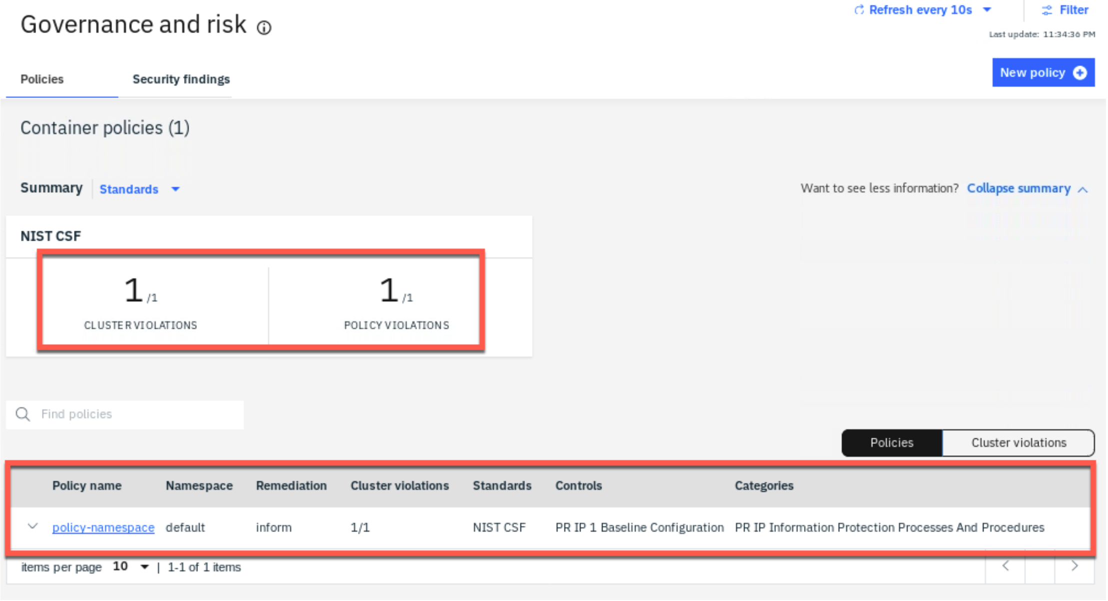
Standards ⓘ
1 × NIST-CSF

Categories ⓘ
1 × PR.IP Information Protection Processes an...

Policy YAML

```yaml
apiVersion: policy.mcm.ibm.com/vlalphal
kind: Policy
metadata:
  name: policy-namespace
  namespace: default
  annotations:
    policy.mcm.ibm.com/standards: NIST-CSF
    policy.mcm.ibm.com/categories: PR.IP Information Protection Processes and Procedures
    policy.mcm.ibm.com/controls: PR.IP-1 Baseline configuration
spec:
  complianceType: musthave
  remediationAction: inform
  disabled: false
  namespaces:
    exclude: ["kube-*"]
    include: ["default"]
  object-templates:
    - complianceType: musthave
      objectDefinition:
        kind: Namespace # must have namespace 'prod'
        apiVersion: v1
        metadata:
          name: k8demo
---
apiVersion: mcm.ibm.com/vlalphal
kind: PlacementBinding
metadata:
  name: binding-policy-namespace
```

__10. In a few seconds, the policy controller will check if the namespace k8demo is present and provides information regarding the current compliance of the policies.
Remember, you did not enforce this policy. Instead we specified inform. As such, the Governance and risk view displays a policy violation in our cluster, as illustrated below.



Governance and risk ⓘ

Container policies (1)

NIST CSF

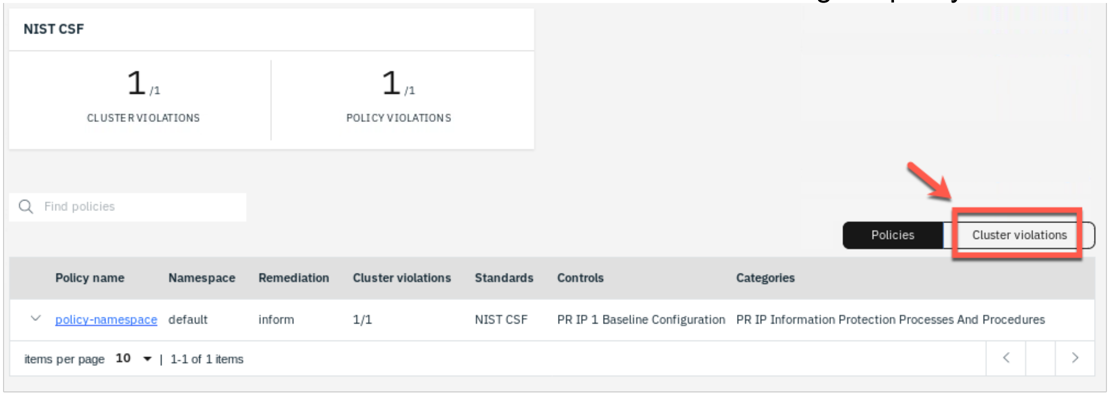
1 /1 CLUSTER VIOLATIONS

1 /1 POLICY VIOLATIONS

Policy name	Namespace	Remediation	Cluster violations	Standards	Controls	Categories
policy-namespace	default	inform	1/1	NIST CSF	PR IP 1 Baseline Configuration	PR IP Information Protection Processes And Procedures

items per page 10 | 1-1 of 1 items

__11. Click the **Cluster violations** link to find which cluster is violating the policy.



NIST CSF

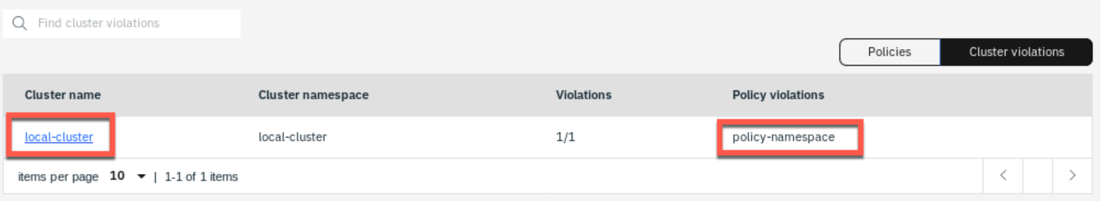
1 /1 CLUSTER VIOLATIONS

1 /1 POLICY VIOLATIONS

Policy name	Namespace	Remediation	Cluster violations	Standards	Controls	Categories
policy-namespace	default	inform	1/1	NIST CSF	PR IP 1 Baseline Configuration	PR IP Information Protection Processes And Procedures

__12. The local-cluster cluster is in violation of the policy which requires a namespace that is called “k8sdemo” to exist.

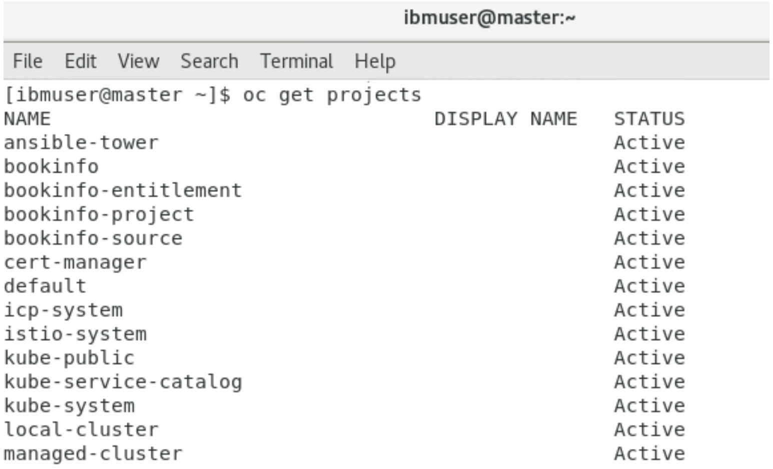
__13. The local-cluster is the same cluster that you verified in the first step of this section that k8sdemo namespace does not exist. Hence it shows that there is no namespace k8demo in the cluster.



Cluster name	Cluster namespace	Violations	Policy violations
local-cluster	local-cluster	1/1	policy-namespace

__14. Now, let’s verify the k8demo namespace still does not exist. Back to the terminal window, run the command below:

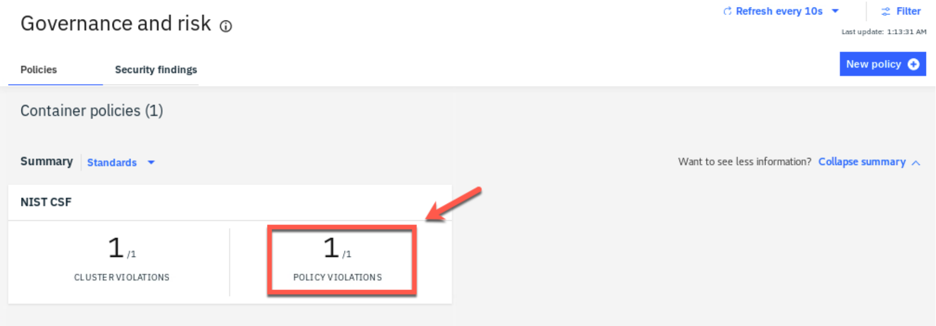
oc get ns



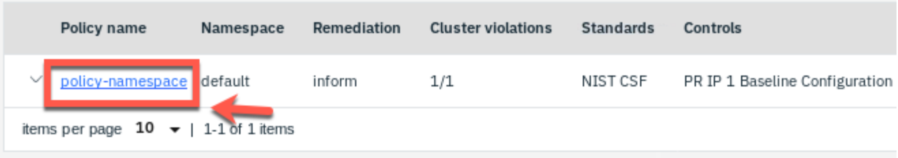
```
[ibmuser@master ~]$ oc get projects
NAME                                DISPLAY NAME   STATUS
ansible-tower                                      Active
bookinfo                                           Active
bookinfo-entitlement                               Active
bookinfo-project                                   Active
bookinfo-source                                    Active
cert-manager                                       Active
default                                            Active
icp-system                                         Active
istio-system                                       Active
kube-public                                        Active
kube-service-catalog                               Active
kube-system                                        Active
local-cluster                                      Active
managed-cluster                                    Active
```

There should not be a namespace named k8demo listed, which indicates that the policy did not enforce it to be created.

__15. Back to Firefox. Now, let’s change the policy to be enforced. In the policies view, click on **POLICY VIOLATIONS**.



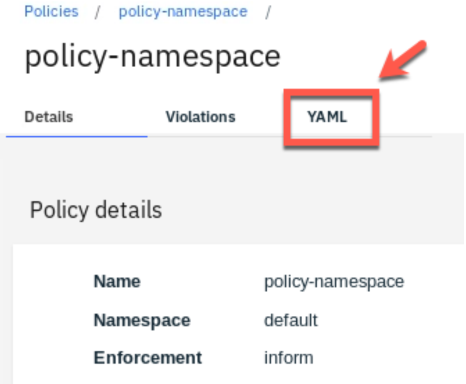
__16. Click the **policy-namespace** link.



Policy name	Namespace	Remediation	Cluster violations	Standards	Controls
policy-namespace	default	inform	1/1	NIST CSF	PR IP 1 Baseline Configuration

items per page 10 | 1-1 of 1 items

__17. Open the **YAML** tab.

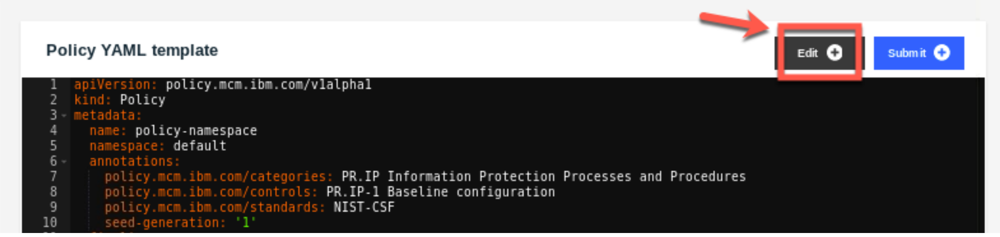


Name policy-namespace

Namespace default

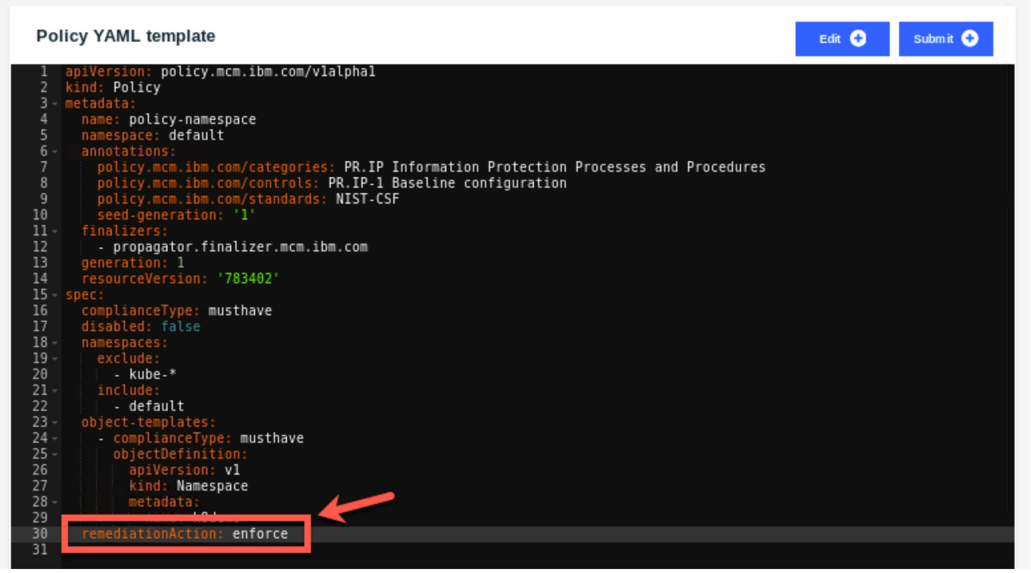
Enforcement inform

__18. Click the Edit button to go into edit mode to modify the YAML file.



```yaml
apiVersion: policy.mcm.ibm.com/vlalphal
kind: Policy
metadata:
  name: policy-namespace
  namespace: default
  annotations:
    policy.mcm.ibm.com/categories: PR.IP Information Protection Processes and Procedures
    policy.mcm.ibm.com/controls: PR.IP-1 Baseline configuration
    policy.mcm.ibm.com/standards: NIST-CSF
    seed-generation: '1'
```

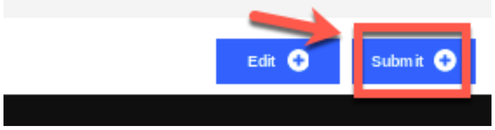
__19. Change the value of remediationAction: inform to **remediationAction: enforce**.



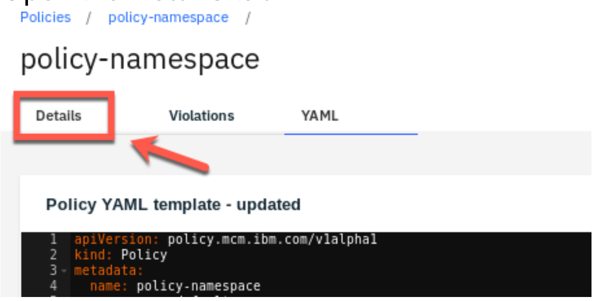
Policy YAML template

```
apiVersion: policy.mcm.ibm.com/vlalphal
kind: Policy
metadata:
  name: policy-namespace
  namespace: default
  annotations:
    policy.mcm.ibm.com/categories: PR.IP Information Protection Processes and Procedures
    policy.mcm.ibm.com/controls: PR.IP-1 Baseline configuration
    policy.mcm.ibm.com/standards: NIST-CSF
    seed-generation: '1'
  finalizers:
    - propagator.finalizer.mcm.ibm.com
  generation: 1
  resourceVersion: '783402'
spec:
  complianceType: musthave
  disabled: false
  namespaces:
    exclude:
      - kube-*
    include:
      - default
  object-templates:
    - complianceType: musthave
      objectDefinition:
        apiVersion: v1
        kind: Namespace
        metadata:
          [illegible]
  remediationAction: enforce
```

__20. Click the **Submit** button to save the change.



__21. Open the **Details** tab.

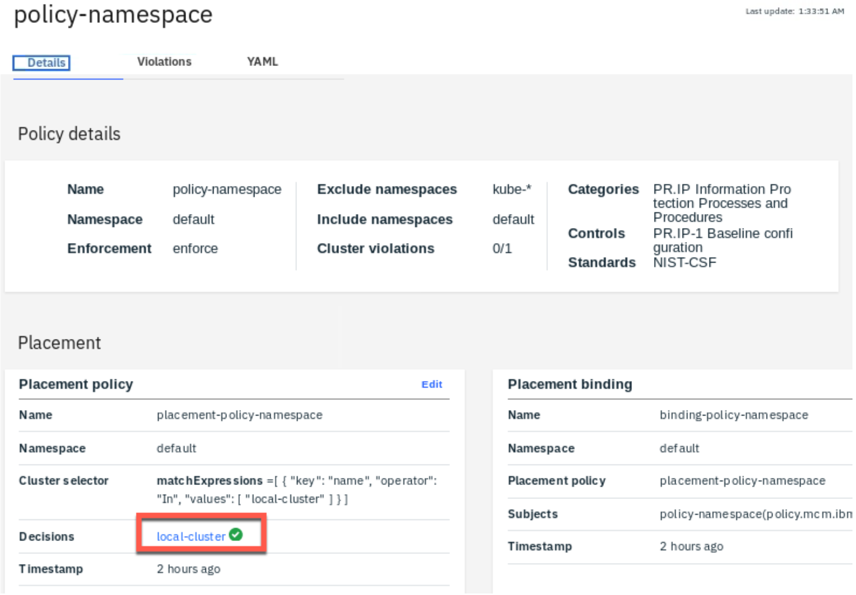


policy-namespace

Policy YAML template - updated

```
apiVersion: policy.mcm.ibm.com/vlalphal
kind: Policy
metadata:
  name: policy-namespace
```

__22. A few seconds later, the policy violation is automatically removed.



policy-namespace

Policy details

Name	policy-namespace	**Exclude namespaces**	kube-*	**Categories**	PR.IP Information Protection Processes and Procedures
Namespace	default	**Include namespaces**	default	**Controls**	PR.IP-1 Baseline configuration
Enforcement	enforce	**Cluster violations**	0/1	**Standards**	NIST-CSF

Placement

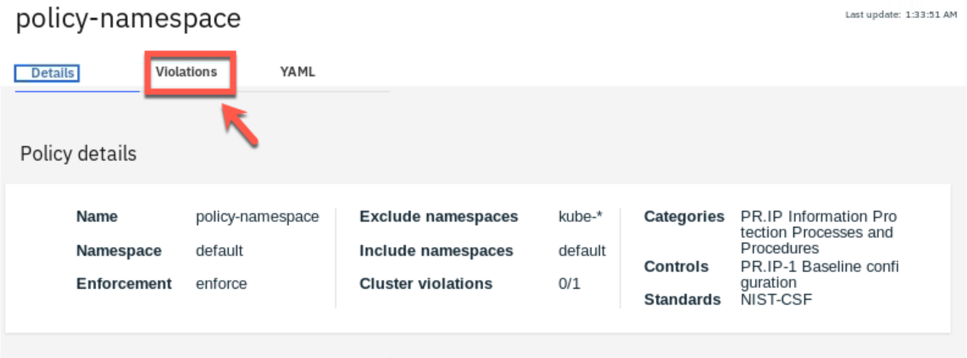
Placement policy

Name	placement-policy-namespace
Namespace	default
Cluster selector	**matchExpressions** =[{ "key": "name", "operator": "In", "values": ["local-cluster"] }]
Decisions	local-cluster ✓
Timestamp	2 hours ago

Placement binding

Name	binding-policy-namespace
Namespace	default
Placement policy	placement-policy-namespace
Subjects	policy-namespace(policy.mcm.ibm
Timestamp	2 hours ago

__23. Open the **Violations** tab.



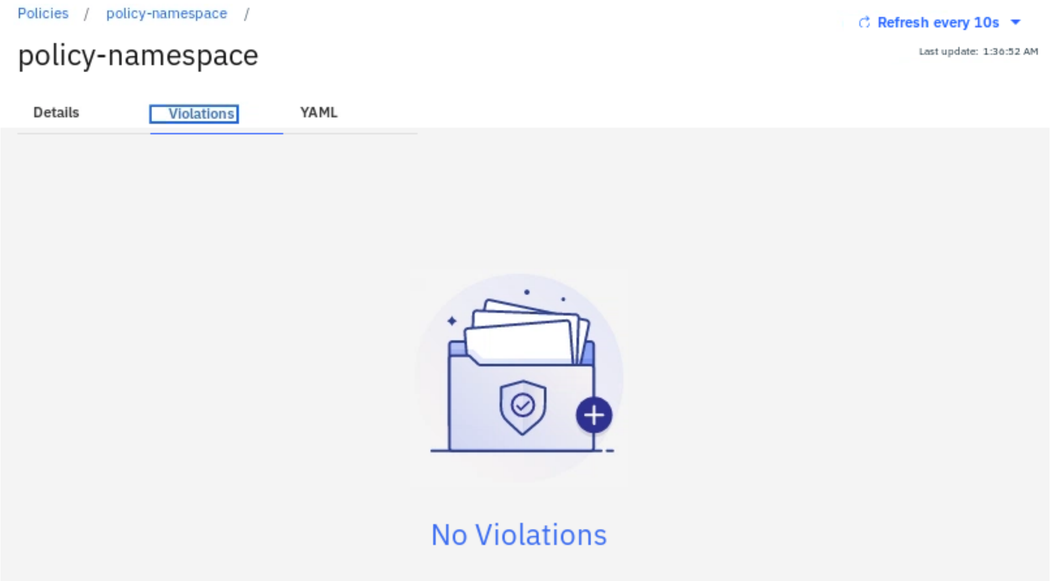
policy-namespace

Last update: 1:33:51 AM

Policy details

Name	policy-namespace	**Exclude namespaces**	kube-*	**Categories**	PR.IP Information Protection Processes and Procedures
Namespace	default	**Include namespaces**	default	**Controls**	PR.IP-1 Baseline configuration
Enforcement	enforce	**Cluster violations**	0/1	**Standards**	NIST-CSF

__24. You also can validate the same from the Violations view: No Violations are available.



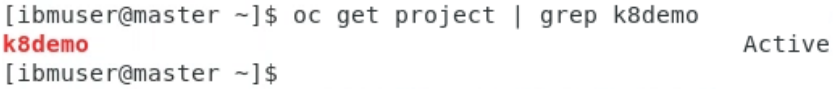
policy-namespace

Last update: 1:36:52 AM

No Violations

__25. Now, let’s check how the policy on enforce mode, removed the violation. Back to the terminal window, run the command below, to ensure that the k8demo namespace is created in the cluster.

oc get project | grep k8demo



```
[ibmuser@master ~]$ oc get project | grep k8demo
k8demo                                           Active
[ibmuser@master ~]$
```

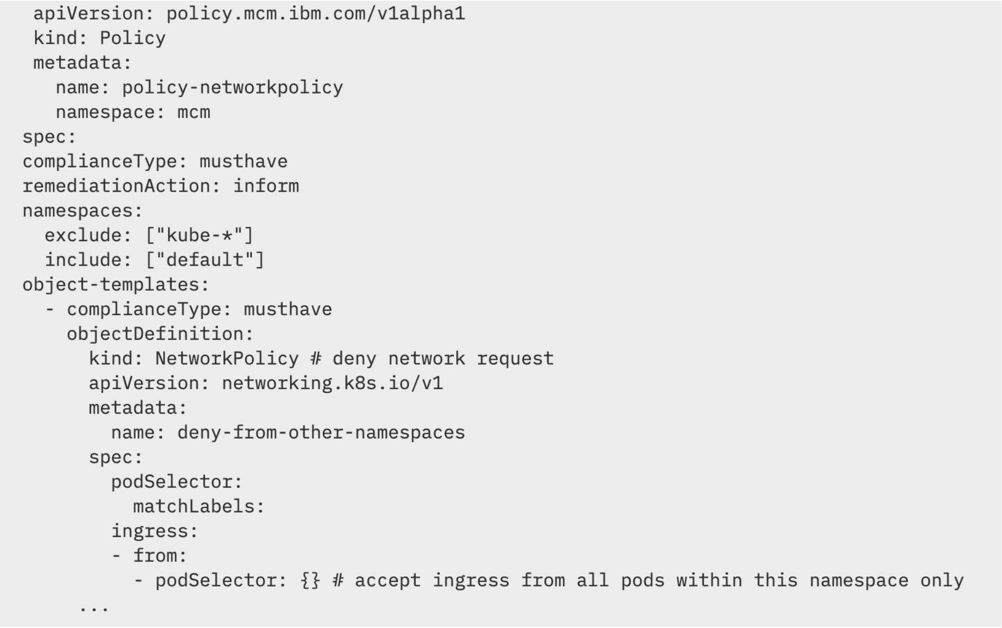
You have successfully implemented the Namespace Policy!

4.3 Create a network policy

A network policy is a specification of how groups of pods are allowed to communicate with each other and other network endpoints.NetworkPolicy resources use labels to select pods and define rules which specify what traffic is allowed to the selected pods.

Apply the network policy to define which network request to deny. For more information about network policies, refer https://kubernetes.io/docs/tasks/administer-cluster/declare-network-policy/

A sample network policy resembles the following:

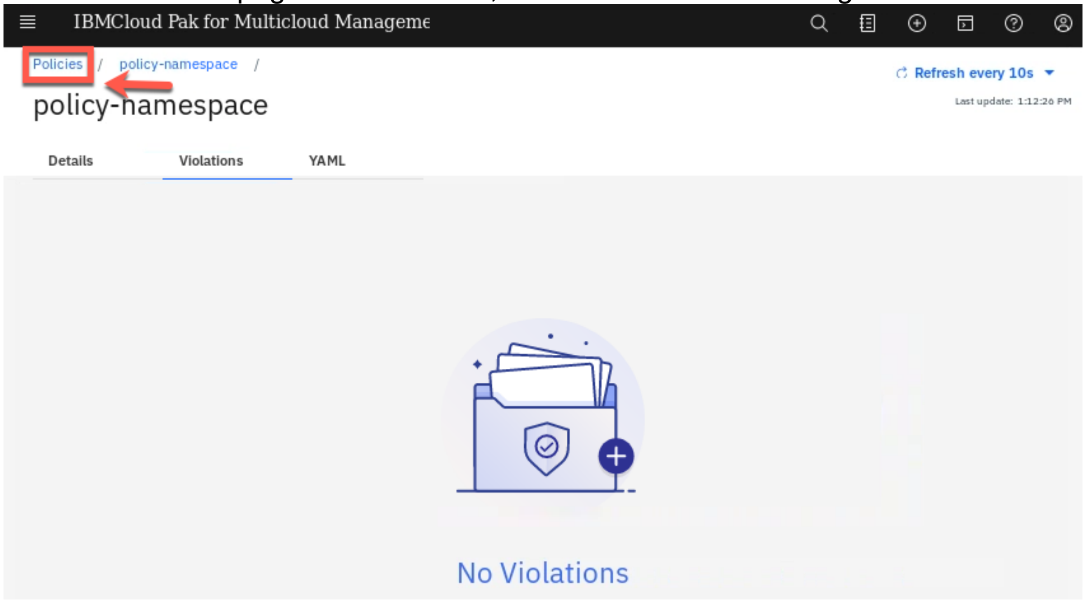


```
 apiVersion: policy.mcm.ibm.com/v1alpha1
 kind: Policy
 metadata:
   name: policy-networkpolicy
   namespace: mcm
spec:
complianceType: musthave
remediationAction: inform
namespaces:
  exclude: ["kube-*"]
  include: ["default"]
object-templates:
  - complianceType: musthave
    objectDefinition:
      kind: NetworkPolicy # deny network request
      apiVersion: networking.k8s.io/v1
      metadata:
        name: deny-from-other-namespaces
      spec:
        podSelector:
          matchLabels:
        ingress:
        - from:
          - podSelector: {} # accept ingress from all pods within this namespace only
     ...
```

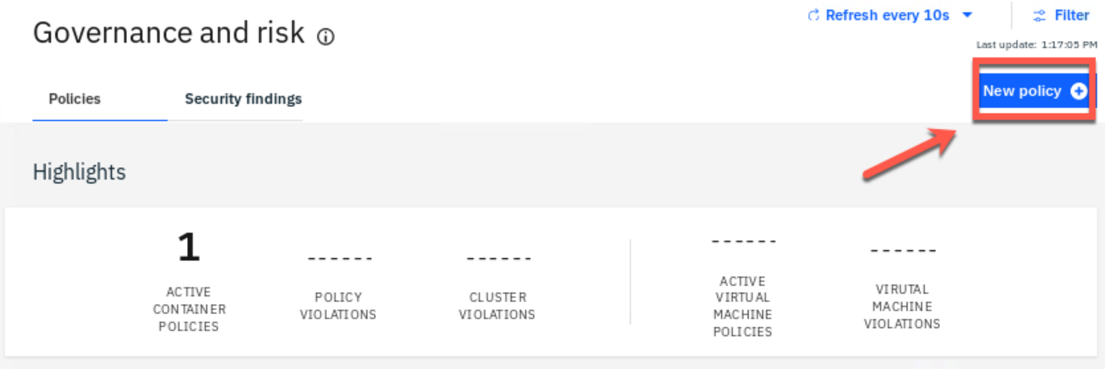
IBM Cloud Pak for Multicloud Management enables your team to check and enforce network policy compliance against your multiple clusters in your hybrid environment.

In this section, you learn how to create a Network Policy in IBM Cloud Pak for Multicloud Management. This lab needs an application that has at least two pods with services where one pod needs to connect to other pod's associated service. You will use the application Quote of the Day that is already deployed on the cluster in the default project.

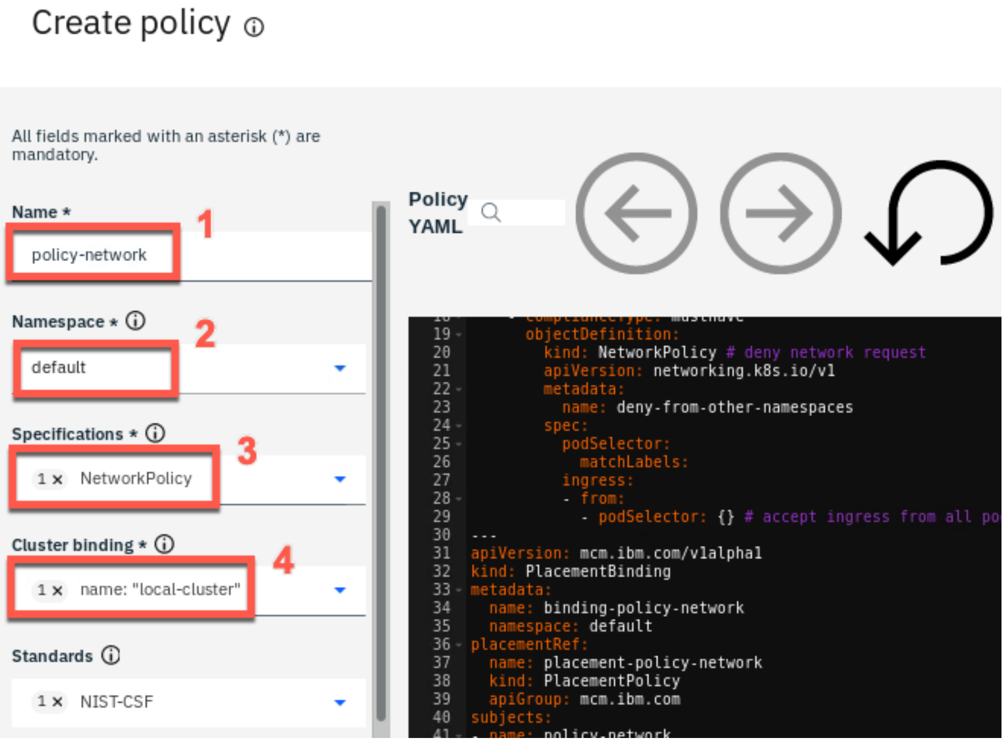
__1. Back to the MCM page tab on Firefox, click **Policies** link on navigation breadcrumb.



__2. On the Policies view, click **New policy**.



__3. Enter **policy-network** as name of the policy (1), select **default** as namespace (2), on specifications field choose **NetworkPolicy – deny network request** (3). On the cluster binding field select **name: “local-cluster”** (4).



Create policy ⓘ

All fields marked with an asterisk (*) are mandatory.

Name * policy-network

Namespace * ⓘ default

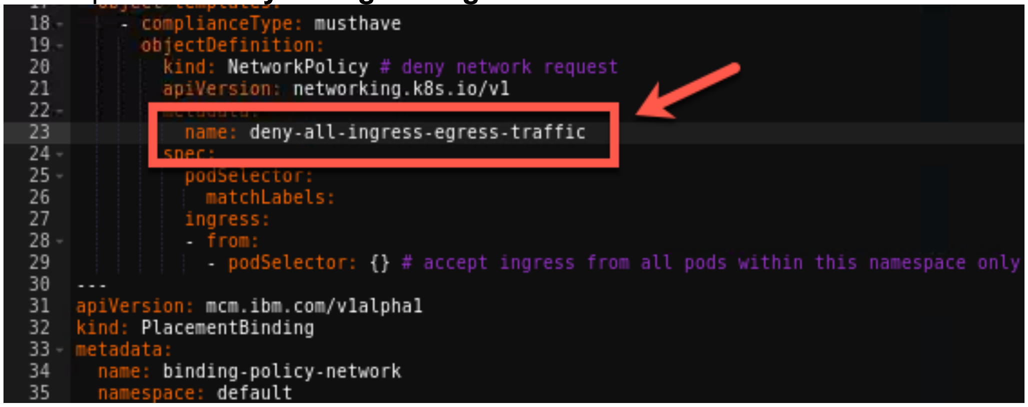
Specifications * ⓘ 1 × NetworkPolicy

Cluster binding * ⓘ 1 × name: "local-cluster"

Standards ⓘ 1 × NIST-CSF

```
    - complianceType: musthave
      objectDefinition:
        kind: NetworkPolicy # deny network request
        apiVersion: networking.k8s.io/v1
        metadata:
          name: deny-from-other-namespaces
        spec:
          podSelector:
            matchLabels:
          ingress:
          - from:
            - podSelector: {} # accept ingress from all po
---
apiVersion: mcm.ibm.com/v1alpha1
kind: PlacementBinding
metadata:
  name: binding-policy-network
  namespace: default
placementRef:
  name: placement-policy-network
  kind: PlacementPolicy
  apiGroup: mcm.ibm.com
subjects:
- name: policy-network
```

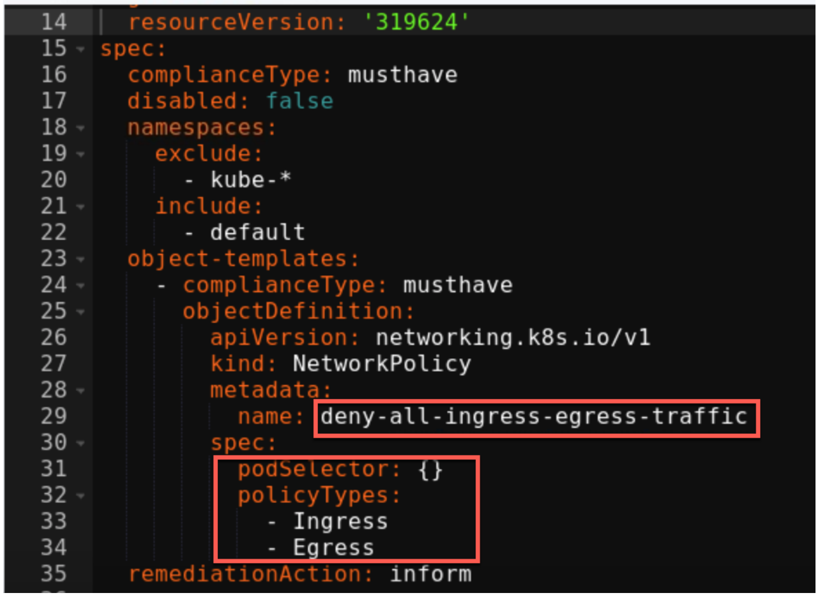
__4. On the YAML file editor, change the objectDefinition > metadata > name from deny-from-other-namespaces to **deny-all-ingress-egress-traffic.**



```
    - complianceType: musthave
      objectDefinition:
        kind: NetworkPolicy # deny network request
        apiVersion: networking.k8s.io/v1
        metadata:
          name: deny-all-ingress-egress-traffic
        spec:
          podSelector:
            matchLabels:
          ingress:
          - from:
            - podSelector: {} # accept ingress from all pods within this namespace only
---
apiVersion: mcm.ibm.com/v1alpha1
kind: PlacementBinding
metadata:
  name: binding-policy-network
  namespace: default
```

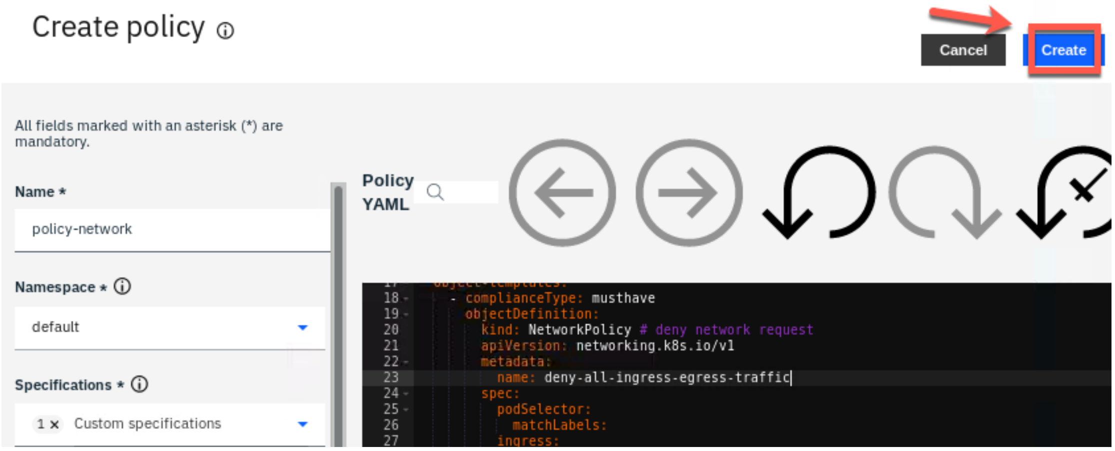
__5. The **spec** section is defined as follow:

spec:
  podSelector: {}
  policyTypes:
    - Ingress
    - Egress



```
  resourceVersion: '319624'
spec:
  complianceType: musthave
  disabled: false
  namespaces:
    exclude:
      - kube-*
    include:
      - default
  object-templates:
    - complianceType: musthave
      objectDefinition:
        apiVersion: networking.k8s.io/v1
        kind: NetworkPolicy
        metadata:
          name: deny-all-ingress-egress-traffic
        spec:
          podSelector: {}
          policyTypes:
            - Ingress
            - Egress
  remediationAction: inform
```

__6. Keep remediationAction as inform by now. Click **Create**.



Create policy ⓘ

All fields marked with an asterisk (*) are mandatory.

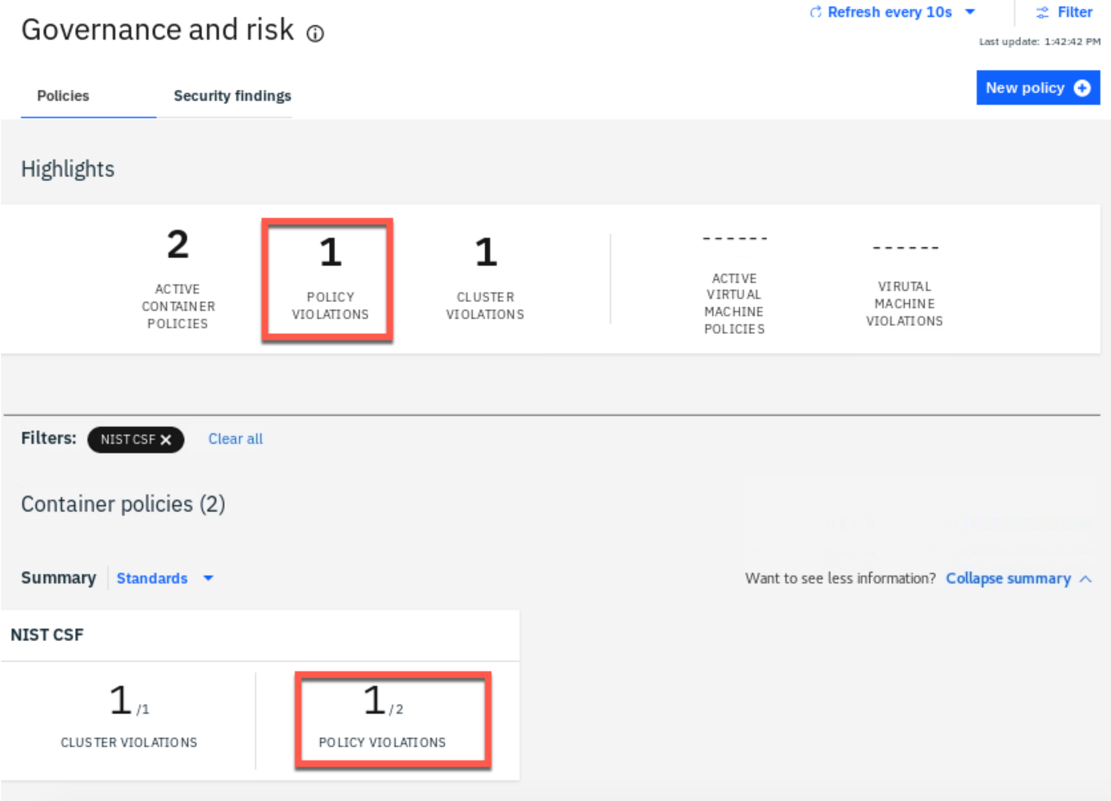
Name * policy-network

Namespace * ⓘ default

Specifications * ⓘ 1 × Custom specifications

```
  object-templates:
    - complianceType: musthave
      objectDefinition:
        kind: NetworkPolicy # deny network request
        apiVersion: networking.k8s.io/v1
        metadata:
          name: deny-all-ingress-egress-traffic
        spec:
          podSelector:
            matchLabels:
          ingress:
```

__7. After few seconds, you should check that the policy violation was detected but not enforced.



Governance and risk ⓘ

Refresh every 10s ▾ | Filter

Last update: 1:42:42 PM

Policies | Security findings | New policy ⊕

Highlights

2	1	1	------	------
ACTIVE CONTAINER POLICIES	POLICY VIOLATIONS	CLUSTER VIOLATIONS	ACTIVE VIRTUAL MACHINE POLICIES	VIRUTAL MACHINE VIOLATIONS

Filters: NIST CSF ✕ Clear all

Container policies (2)

Summary | Standards ▾

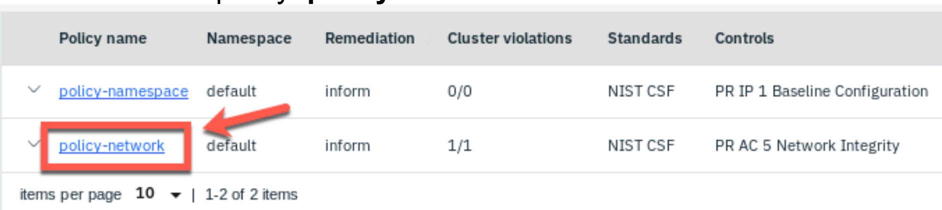
Want to see less information? Collapse summary ^

NIST CSF

1 /1	1 /2
CLUSTER VIOLATIONS	POLICY VIOLATIONS

Because the policy is on inform mode, the policy is not forced.

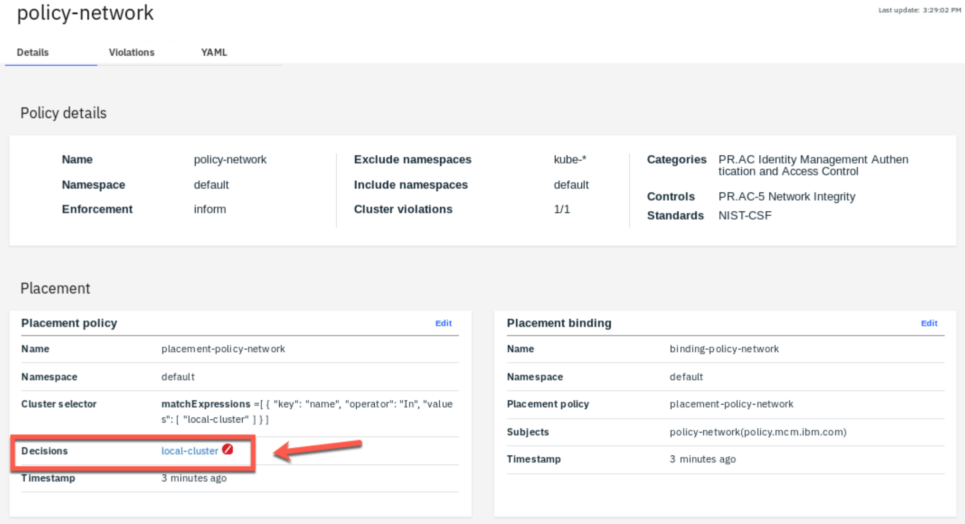
__8. Click on the new policy: **policy-network**.



Policy name	Namespace	Remediation	Cluster violations	Standards	Controls
policy-namespace	default	inform	0/0	NIST CSF	PR IP 1 Baseline Configuration
policy-network	default	inform	1/1	NIST CSF	PR AC 5 Network Integrity

items per page 10 ▾ | 1-2 of 2 items

__9. You see that one decision is in violation.



policy-network

Last update: 3:29:02 PM

Details | Violations | YAML

Policy details

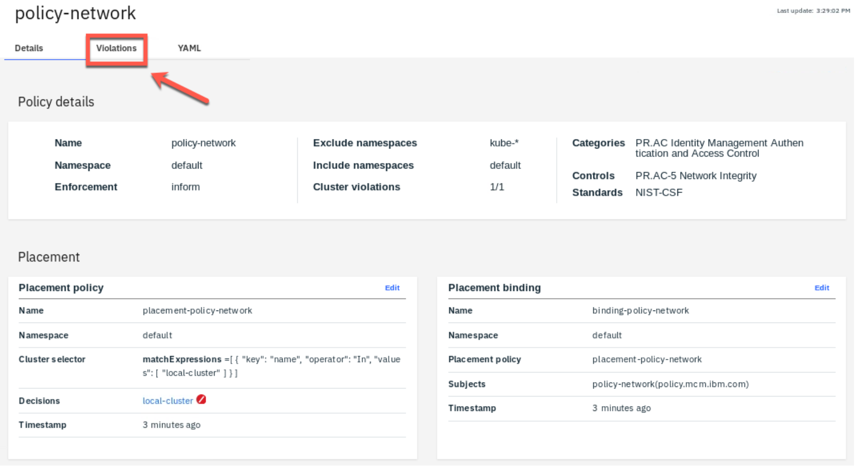
Name	policy-network	Exclude namespaces	kube-*	Categories	PR.AC Identity Management Authen tication and Access Control
Namespace	default	Include namespaces	default	Controls	PR.AC-5 Network Integrity
Enforcement	inform	Cluster violations	1/1	Standards	NIST-CSF

Placement

Placement policy	Edit
Name	placement-policy-network
Namespace	default
Cluster selector	**matchExpressions** =[{ "key": "name", "operator": "In", "values": ["local-cluster"] }]
Decisions	local-cluster ⊘
Timestamp	3 minutes ago

Placement binding	Edit
Name	binding-policy-network
Namespace	default
Placement policy	placement-policy-network
Subjects	policy-network(policy.mcm.ibm.com)
Timestamp	3 minutes ago

__10. Click the **Violations** tab.



policy-network

Last update: 3:29:02 PM

Details | Violations | YAML

Policy details

Name	policy-network	**Exclude namespaces**	kube-*	**Categories**	PR.AC Identity Management Authentication and Access Control
Namespace	default	**Include namespaces**	default	**Controls**	PR.AC-5 Network Integrity
Enforcement	inform	**Cluster violations**	1/1	**Standards**	NIST-CSF

Placement

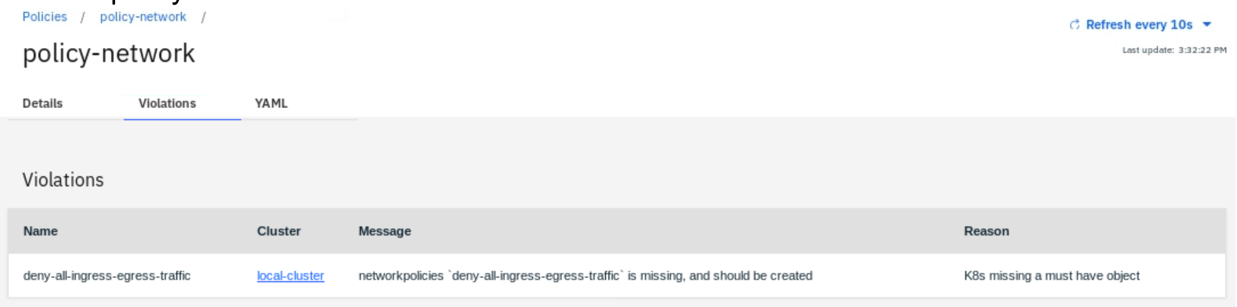
Placement policy Edit

Name	placement-policy-network
Namespace	default
Cluster selector	**matchExpressions** =[{ "key": "name", "operator": "In", "values": ["local-cluster"] }]
Decisions	local-cluster
Timestamp	3 minutes ago

Placement binding Edit

Name	binding-policy-network
Namespace	default
Placement policy	placement-policy-network
Subjects	policy-network(policy.mcm.ibm.com)
Timestamp	3 minutes ago

__11. Here you see the violation about network isolation. Your local-cluster is not compliance with this network policy. Let's fix it!



Policies / policy-network /

policy-network

Refresh every 10s

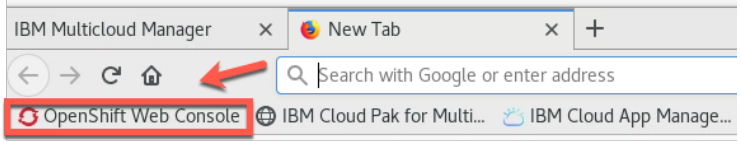
Last update: 3:32:22 PM

Details | Violations | YAML

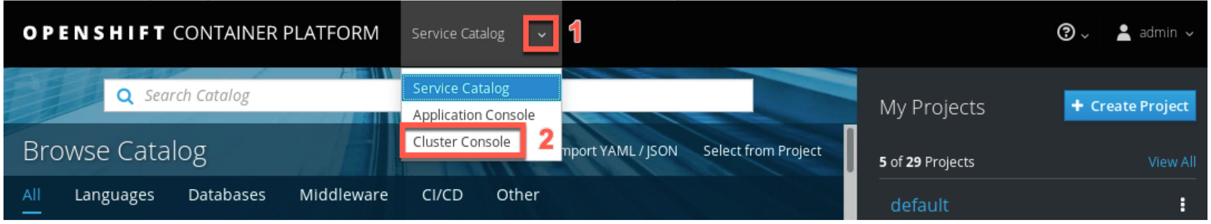
Violations

Name	Cluster	Message	Reason
deny-all-ingress-egress-traffic	local-cluster	networkpolicies `deny-all-ingress-egress-traffic` is missing, and should be created	K8s missing a must have object

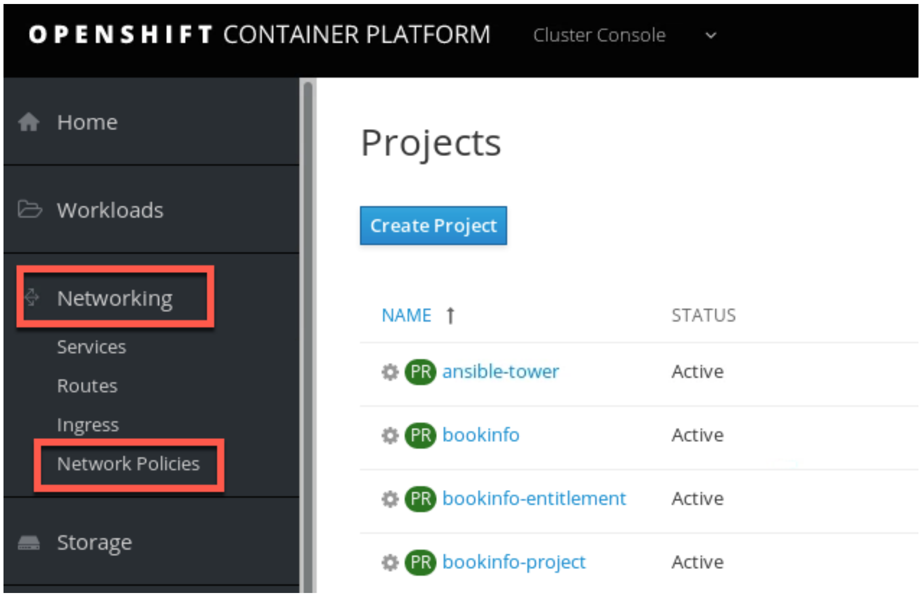
__12. Open a new Firefox browser, and open the **OpenShift Web Console** (there is a link on Bookmark bar).



__13. If necessary, log in with admin/Passw0rd! and change to **Cluster Console** view.

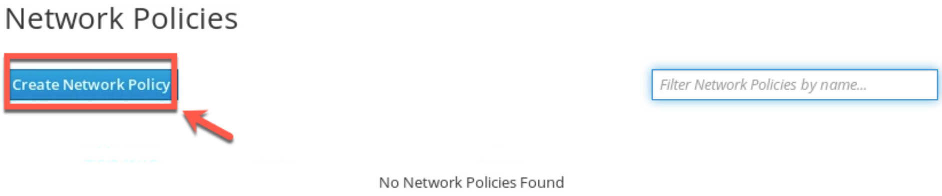


__14. You should see the projects page. Use the side bar menu and select **Networking > Network Policies**.



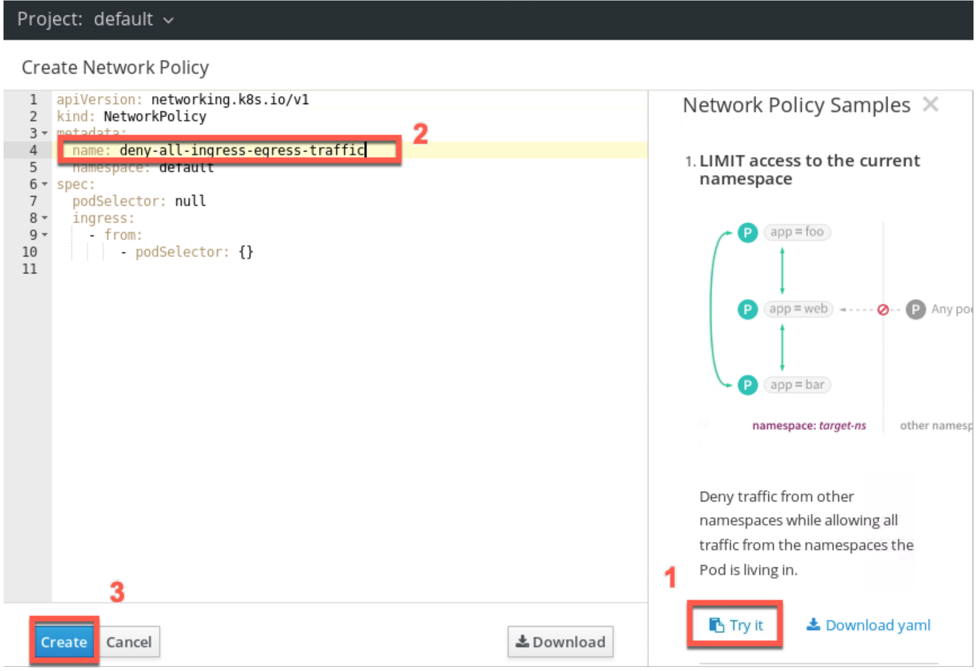
NAME ↑	STATUS
PR ansible-tower	Active
PR bookinfo	Active
PR bookinfo-entitlement	Active
PR bookinfo-project	Active

__15. Click the **Create Network Policy** button.



No Network Policies Found

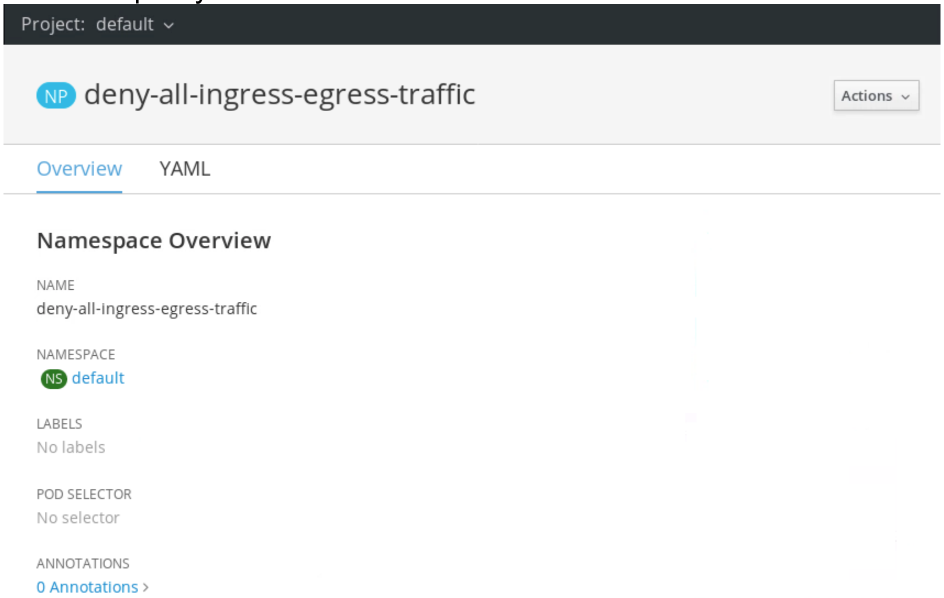
__16. To the right of the view are a list of policy samples. Fortunately for us, there is a Network Policy Sample that we can use. Click the **Try it** link (1). Then edit the YAML and insert the letters from to the metadata name, replace deny-other-namespaces to **deny-all-ingress-egress-traffic** (2). Click the **Create** button to create this policy (3).



```
apiVersion: networking.k8s.io/v1
kind: NetworkPolicy
metadata:
  name: deny-all-ingress-egress-traffic
  namespace: default
spec:
  podSelector: null
  ingress:
    - from:
        - podSelector: {}
```

Deny traffic from other namespaces while allowing all traffic from the namespaces the Pod is living in.

__17. Great the policy is created now.



Project: default

NP deny-all-ingress-egress-traffic

Actions

Overview YAML

Namespace Overview

NAME
deny-all-ingress-egress-traffic

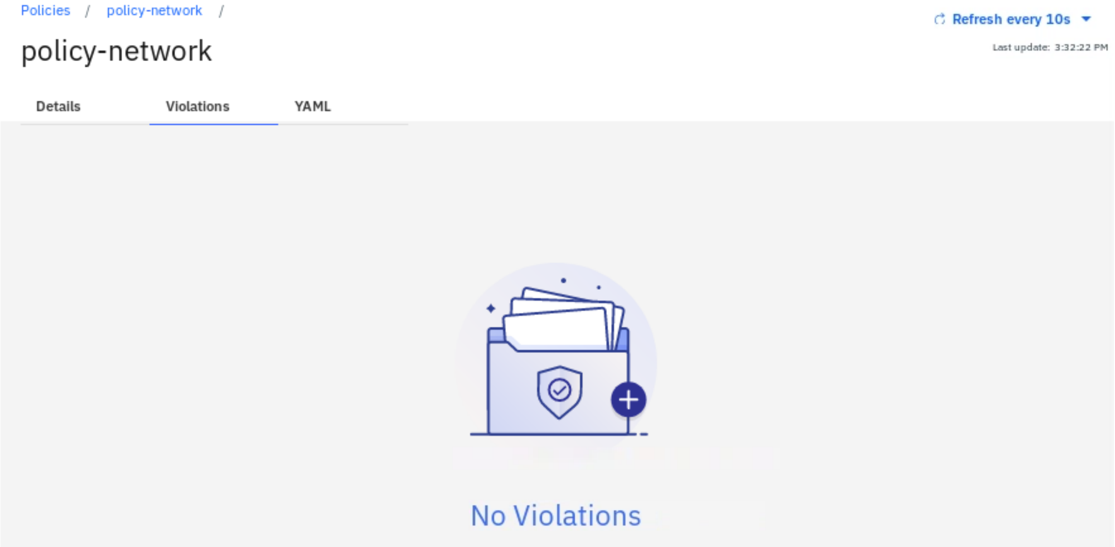
NAMESPACE
NS default

LABELS
No labels

POD SELECTOR
No selector

ANNOTATIONS
0 Annotations >

__18. Go back to the MCM browser tab. You should notice that there are no violations now (wait 10 seconds if you have to).



Policies / policy-network /

Refresh every 10s

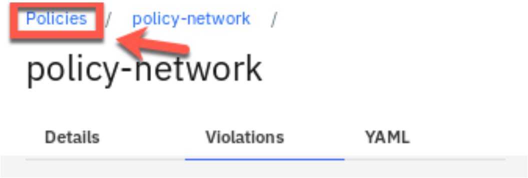
Last update: 3:32:22 PM

policy-network

Details Violations YAML

No Violations

__19. Click the **Policies** link in breadcrumbs, to go back to the Policies page.

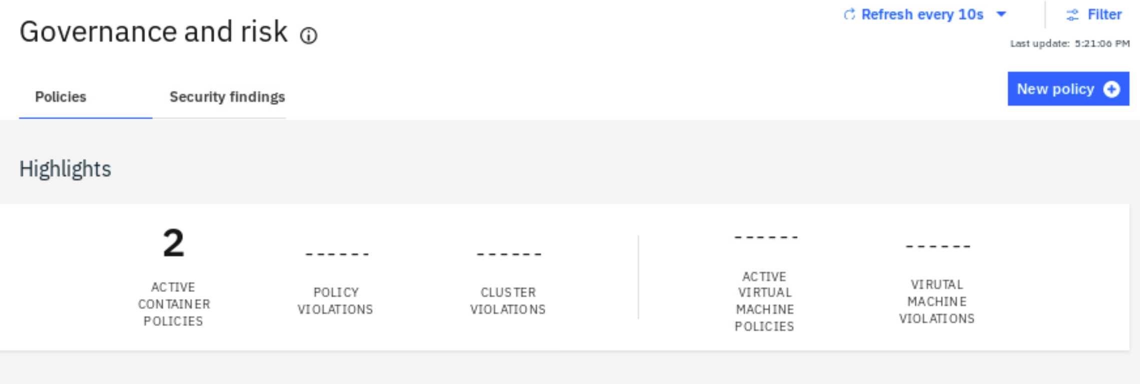


Policies / policy-network /

policy-network

Details Violations YAML

__20. After waiting 10 seconds, you should see that all violations are gone.



Great! You have successfully implemented Network Policy!

Congratulations! You have successfully completed the lab “Security and Compliance Management with IBM Cloud Pak for Multicloud Management”.

Summary

IBM Cloud Pak for Multicloud Management governance and risk policy framework helps create custom policy controllers. You learned in the Lab how to create and customize policies with the out of the box policy templates.   If you would like to learn more about Cloud Pak for Multicloud Management, refer to:

- Cloud Pak for Multicloud Management home page
- Cloud Pak for Multicloud Management Demos

Appendix A. Notices

This information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

IBM World Trade Asia Corporation
Licensing
2-31 Roppongi 3-chome, Minato-ku
Tokyo 106-0032, Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

All statements regarding IBM's future direction and intent are subject to change or withdrawal without notice, and represent goals and objectives only.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental. All references to fictitious companies or individuals are used for illustration purposes only.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs.

Appendix B. Trademarks and copyrights

The following terms are trademarks of International Business Machines Corporation in the United States, other countries, or both:

IBM	AIX	CICS	ClearCase	ClearQuest	Cloudscape
Cube Views	DB2	developerWorks	DRDA	IMS	IMS/ESA
Informix	Lotus	Lotus Workflow	MQSeries	OmniFind	
Rational	Redbooks	Red Brick	RequisitePro	StrongLoop	System i
System z	*Tivoli*	*WebSphere*	*Workplace*	*System p*	

Adobe, the Adobe logo, PostScript, and the PostScript logo are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States, and/or other countries.

IT Infrastructure Library is a registered trademark of the Central Computer and Telecommunications Agency which is now part of the Office of Government Commerce.

Intel, Intel logo, Intel Inside, Intel Inside logo, Intel Centrino, Intel Centrino logo, Celeron, Intel Xeon, Intel SpeedStep, Itanium, and Pentium are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

ITIL is a registered trademark, and a registered community trademark of The Minister for the Cabinet Office, and is registered in the U.S. Patent and Trademark Office.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Java and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.

Cell Broadband Engine is a trademark of Sony Computer Entertainment, Inc. in the United States, other countries, or both and is used under license therefrom.

Linear Tape-Open, LTO, the LTO Logo, Ultrium, and the Ultrium logo are trademarks of HP, IBM Corp. and Quantum in the U.S. and other countries.

NOTES

NOTES