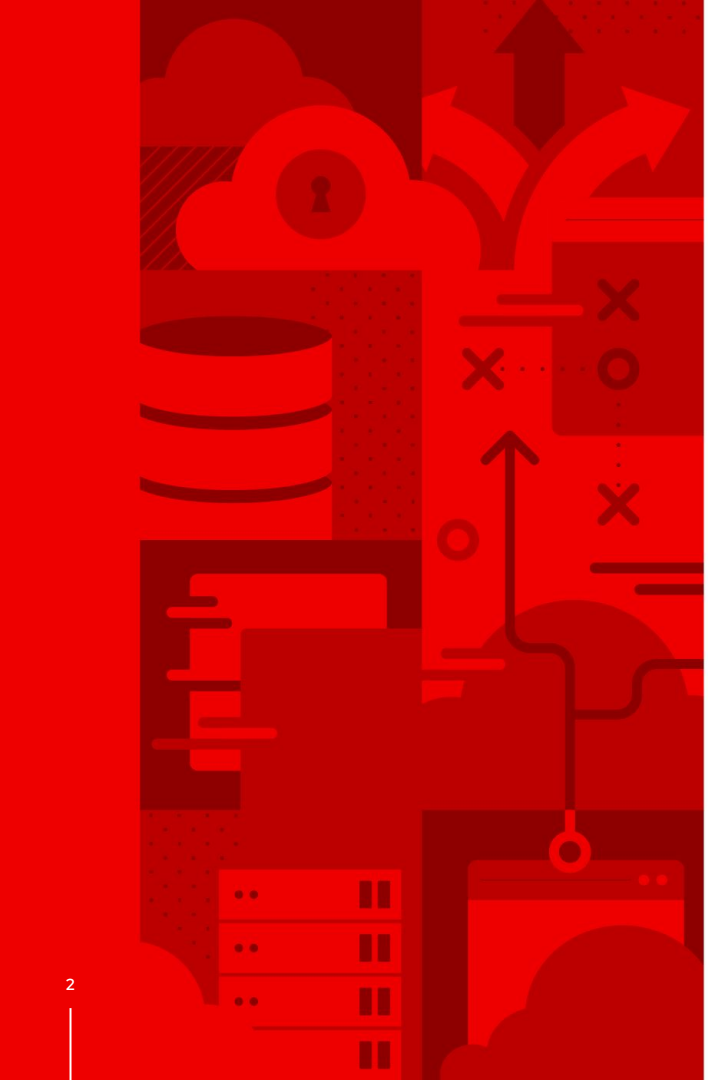
Overview of Cloud Native Functions Certification Framework

Shimrit Peretz

Red Hat



Overview

Red Hat

Process

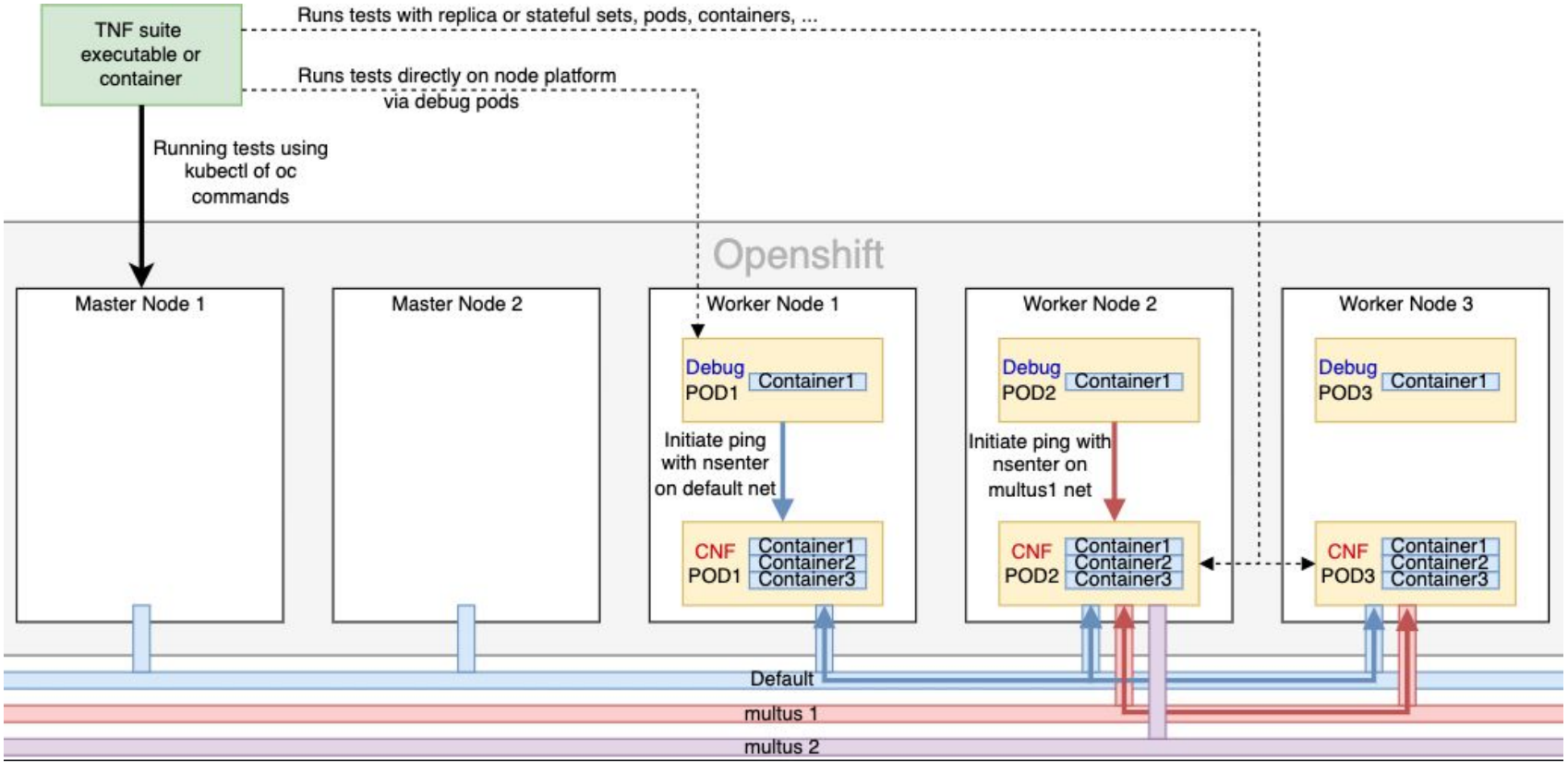
Red Hat

- This repository provides a set of Cloud-Native Network Function (CNF) test cases and the framework to add more test cases.

- The tests and framework are intended to test the interaction of CNFs with OpenShift Container Platform. It also generates a report (claim.json) after completing the tests.

- CATALOG.md provides a list of test cases and test case building blocks included in the test suite.

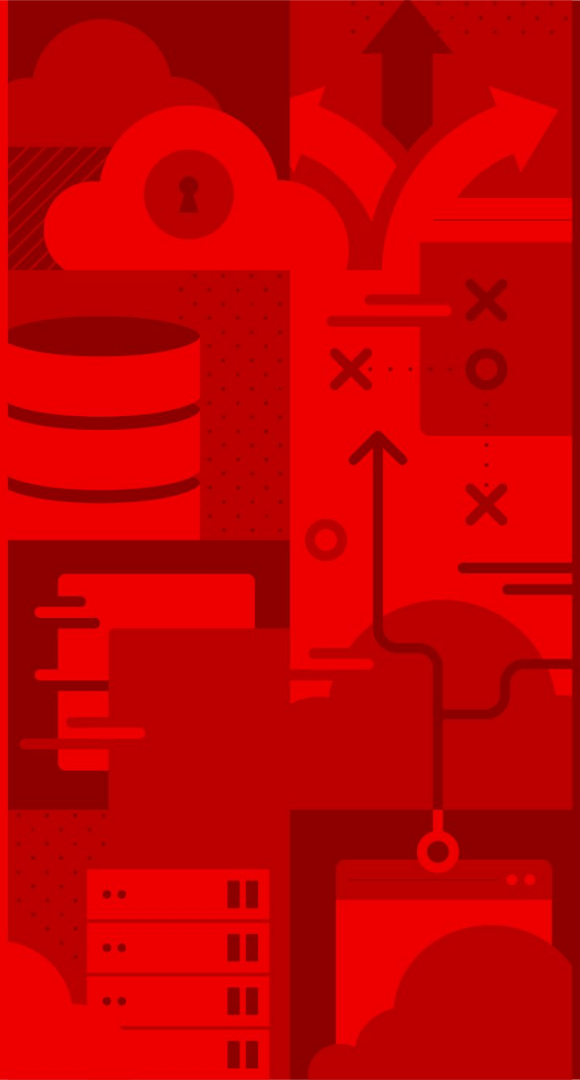
Architecture



Included Test Suites

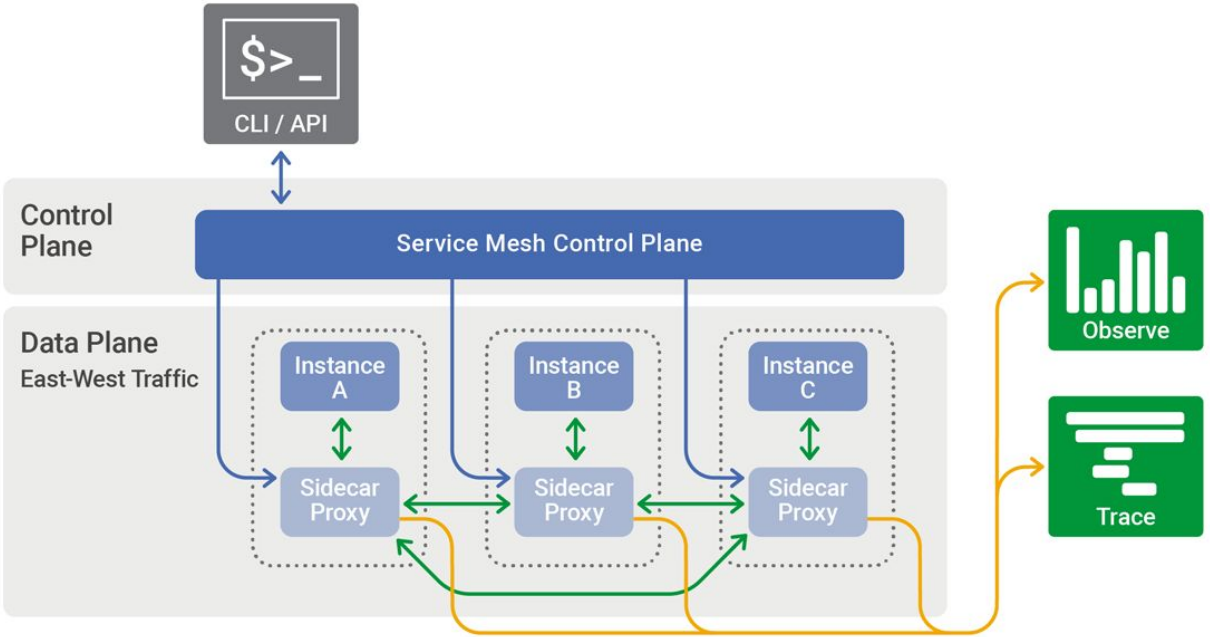
- access-control
- affiliated-certification
- operator
- lifecycle
- observability
- platform -alteration
- networking

Service mesh



Observability in highly dynamic microservices environment

Microservices observability



Red Hat

What might help to achieve this level of observability?

- Infrastructure layer that aids in communication between services or microservices, using a proxy.

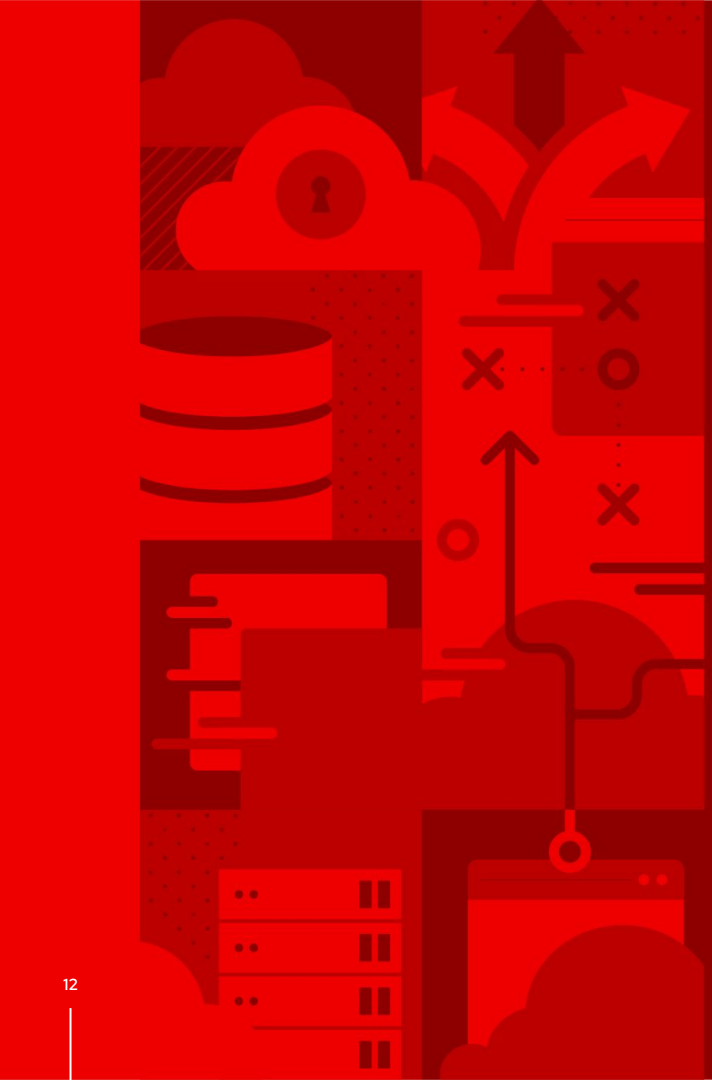
- As applications are decomposed from monoliths, all of the resulting microservices need new tools to address the connectivity challenges that arise in handling distributed services.

- An Istio service mesh defines both the control plane and the data plane.

Service Mesh

- A service mesh, like the open source project Istio, is a way to control how different parts of an application share data with one another.

- A service mesh is a dedicated infrastructure layer built right into an app.



What you need for Service mesh?

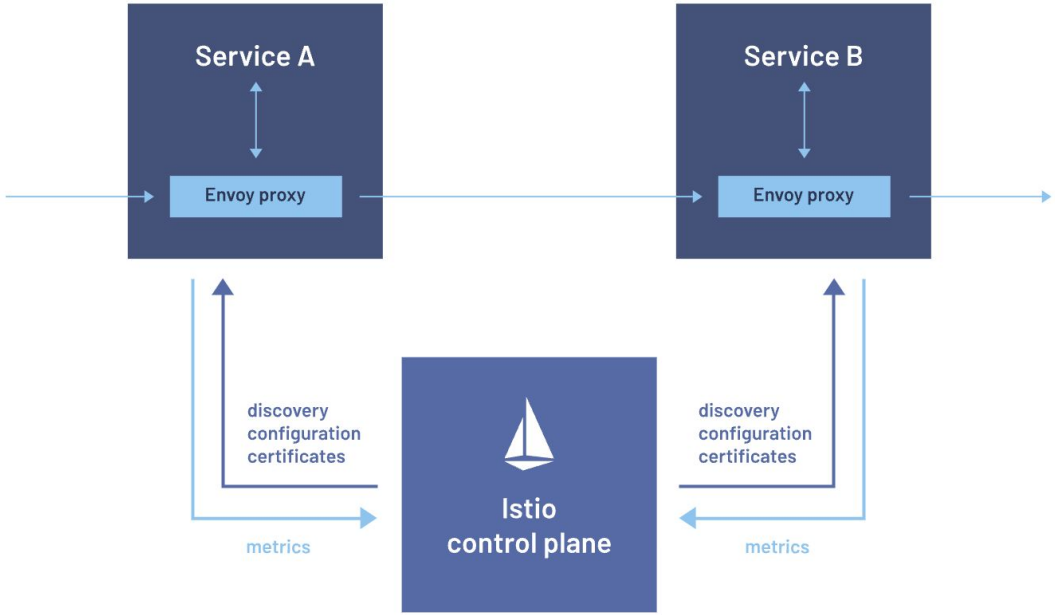
Install Istio

```
shimritperetz@speretz-mac istio-1.14.1 % istioctl install --set profile=demo -y

✔ Istio core installed
✔ Istiod installed
✔ Egress gateways installed
✔ Ingress gateways installed
✔ Installation complete
Making this installation the default for injection and validation.

Thank you for installing Istio 1.14.  Please take a few minutes to tell us about your install/upgrade experience!  https://forms.gle/y
EtCbt45FZ3VoDT5A
shimritperetz@speretz-mac istio-1.14.1 % oc get ns
NAME                 STATUS   AGE
default              Active   27d
istio-system         Active   5m46s
kube-node-lease      Active   27d
kube-public          Active   27d
kube-system          Active   27d
local-path-storage   Active   27d
tnf                  Active   27d
shimritperetz@speretz-mac istio-1.14.1 % oc label namespace tnf istio-injection=enabled
```

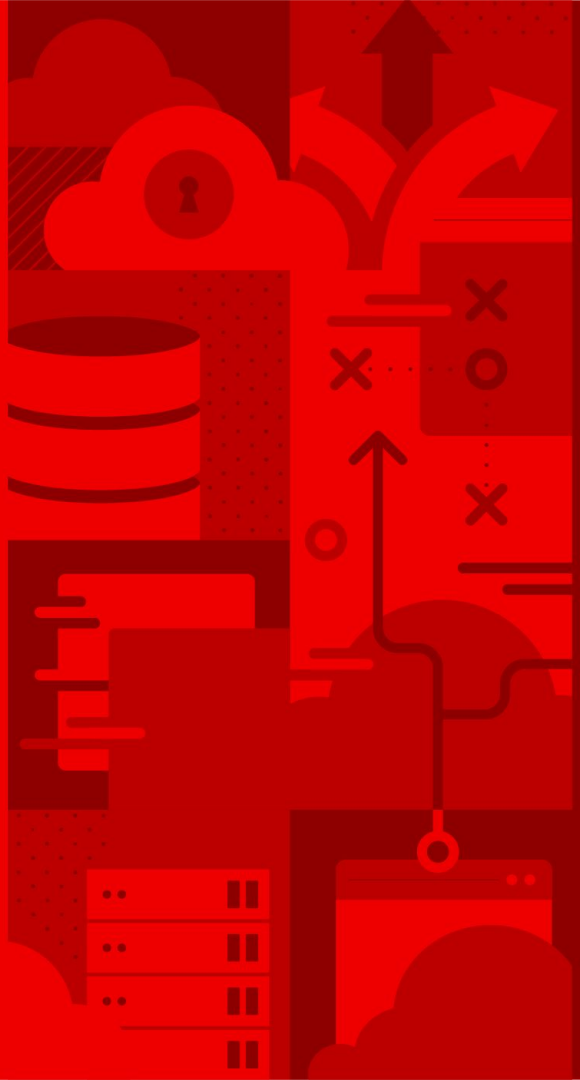
Istio

- TLS encrypted communication, identity-based authentication & authorization.
- Load balancing (HTTP, gRPC, WebSocket, TCP traffic)
- Traffic control (with rich routing rules, retries, failovers, fault injection)
- Policy layer & configuration API (for access controls, rate limits & quotas)
- Auto metrics, logs, traces (for all ingress & egress traffic in a cluster)

How it works?



Red Hat



The Code

Red Hat

```go
func TestServiceMesh(env *provider.TestEnvironment) {
    // check if istio is installed
    if !env.IstioServiceMesh {
        tnf.ClaimFilePrintf("Istio is not installed")
        return
    }
    tnf.ClaimFilePrintf("Istio is installed")
```

In file autodiscover.go:

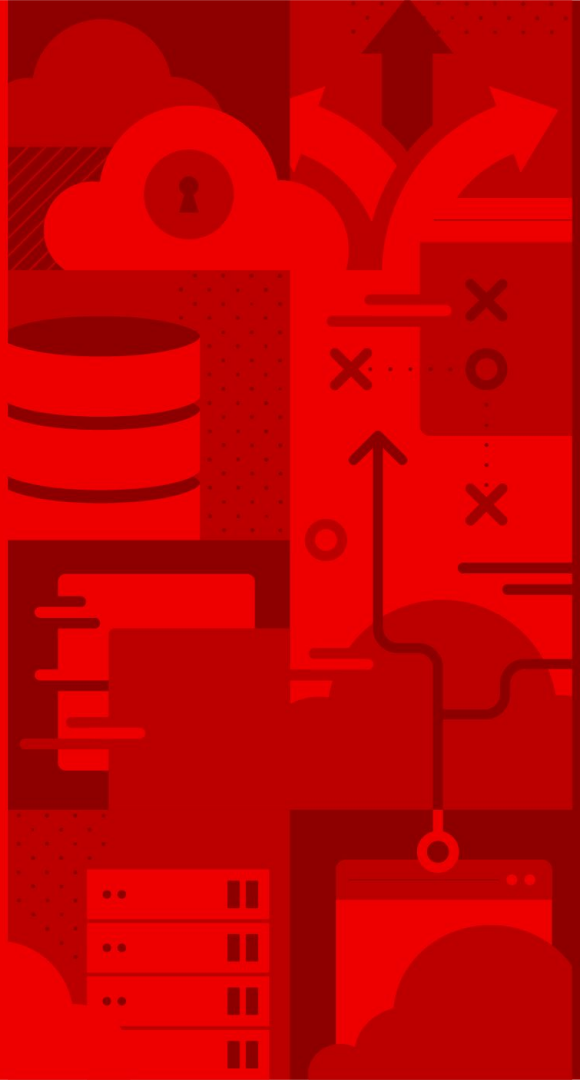
```go
    data.Istio = findnamespace(oc.K8sClient.CoreV1())
```

In file autodiscover_operators.go:

```go
func findnamespace(oc corev1client.CoreV1Interface) bool {
    nsList, err := oc.Namespaces().List(context.TODO(), metav1.ListOptions{})
    if err != nil {
        logrus.Errorln("Error when listing", "err: ", err)
    }
    for index := range nsList.Items {
        if nsList.Items[index].ObjectMeta.Name == istio {
            return true
        }
    }
    return false
}
```

In file suite.go in Platform:

```
137     var badPods []string
138     for _, put := range env.Pods {
139         for _, cut := range put.Containers {
140             if cut.Status.Name == istio {
141                 tnf.ClaimFilePrintf("For pods %s ,ns %s have service mesh", cut.Podname, cut.Namespace)
142             } else {
143                 badPods = append(badPods, "pod "+cut.Podname+" ,ns "+cut.Namespace+" do not have service mesh")
144             }
145         }
146     }
147     logrus.Println("bad pods ", badPods)
148 }
149
```



Try it out yourself

https://github.com/test-network-function/cnf-certification-test/pull/228

https://github.com/test-network-function/cnf-certification-test/pull/291/files

Red Hat