

Configuring Networking for z/OS Container Platform

SHARE 2026 Winter Technical Conference
Session 34085 – February 26, 2026

Ben Hicks - ben.hicks@ibm.com

Mary Julian - Mary.Julian@ibm.com

Agenda

- Quick Overview of Containers and Kubernetes
- What is z/OS Container Platform (zOSCP)
- Configuring your network for Podman
- Building and running your first container image - demo
- Multiple VIPARANGE statements for zOSCP
- Configuring your network for Kubernetes
- Initializing your Kubernetes cluster
- Deploying your container image into a Kubernetes cluster
- High Availability for your Kubernetes cluster

Overview of containers and Kubernetes

Containers overview

- Standard units of software that package up application code and required dependencies
- Containers are operating system and HW-architecture specific
- Utilize facilities provided by the underlying operating system to run applications in isolated environments
- Make moving, stacking, unstacking of compliant software easier – provides for application portability and isolation

Kubernetes overview

Open-source container orchestration framework

Helps manage containerized applications

- The trend for application development is to break monolithic applications into many microservices
 - Containers provide the perfect environment for running microservices
- Orchestration becomes important when there are many application instances (microservices) across many systems/environments spanning physical machines, VMs, cloud

Features

- Framework for deploying microservices
- Automatic rollouts and rollbacks
- Service discovery and load balancing
- Horizontal scaling
- Self-healing
- Secret and configuration management

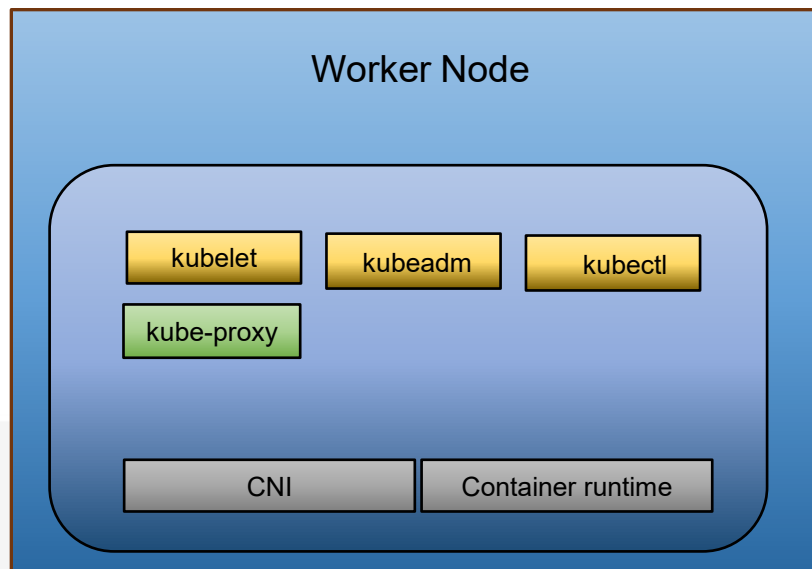
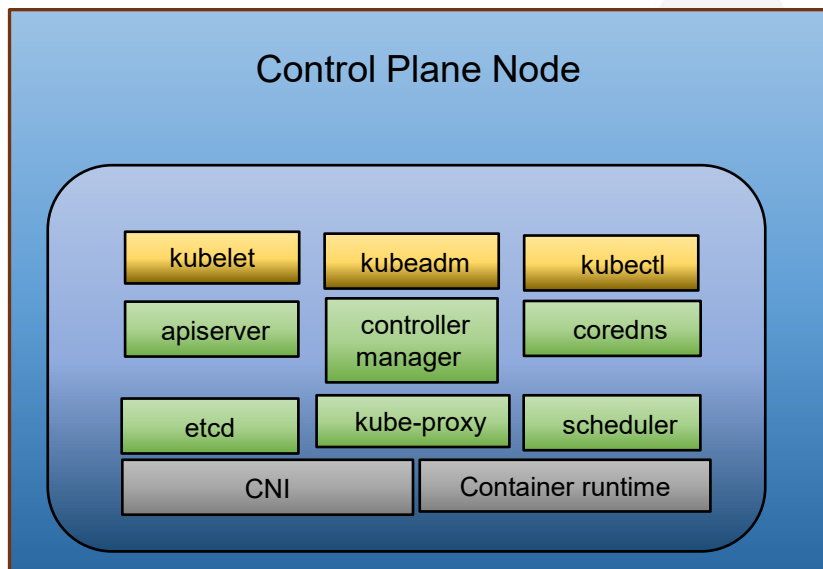
Kubernetes architecture

Control Plane Nodes

- Manage and provide the infrastructure for the Kubernetes cluster

Worker Nodes

- Host the container applications (microservices)



- Utility or Daemon
- Container
- Not part of Kubernetes

What is z/OS Container Platform (zOSCP)?

z/OS Container Platform positioning



z/OS Container Extension (zCX)

- Containerized Linux on Z applications running on z/OS
- Linux applications requiring low latency access to z/OS apps/data
- Linux virtual appliance running as part of z/OS



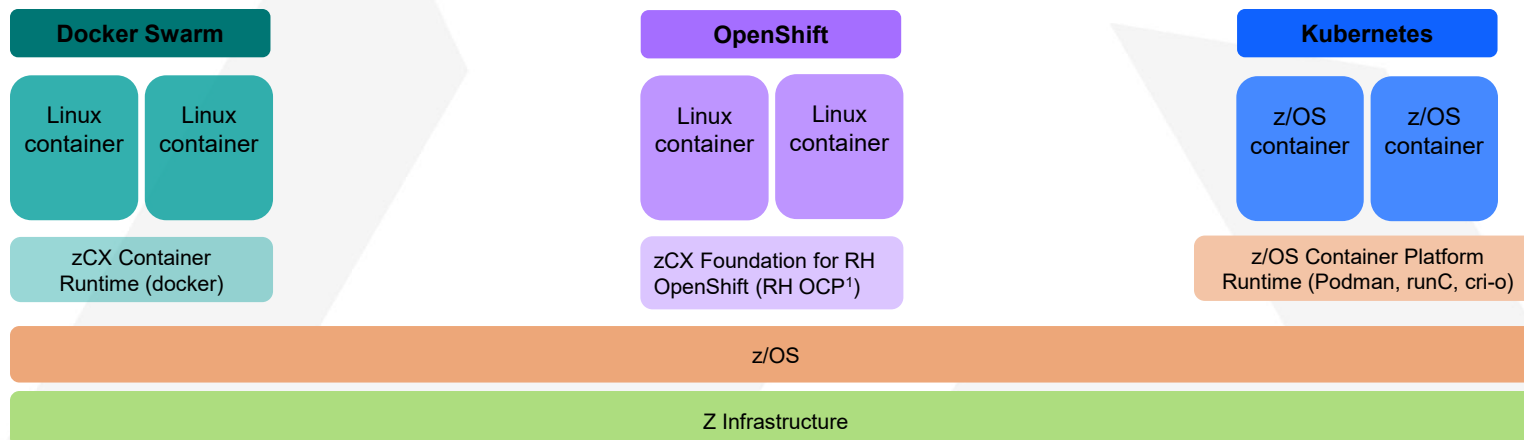
zCX Foundation for OpenShift

- Containerized Linux on Z application running in Red Hat OpenShift on z/OS
- Co-locating applications with non-containerized workloads/data on z/OS
- Red Hat CoreOS included with RH OpenShift



IBM z/OS Container Platform

- Containerized z/OS UNIX applications running on z/OS
- All cluster components run on z/OS



¹ OpenShift Container Platform includes Red Hat CoreOS, cri-o, Kubernetes, OpenShift

z/OS Containers product components

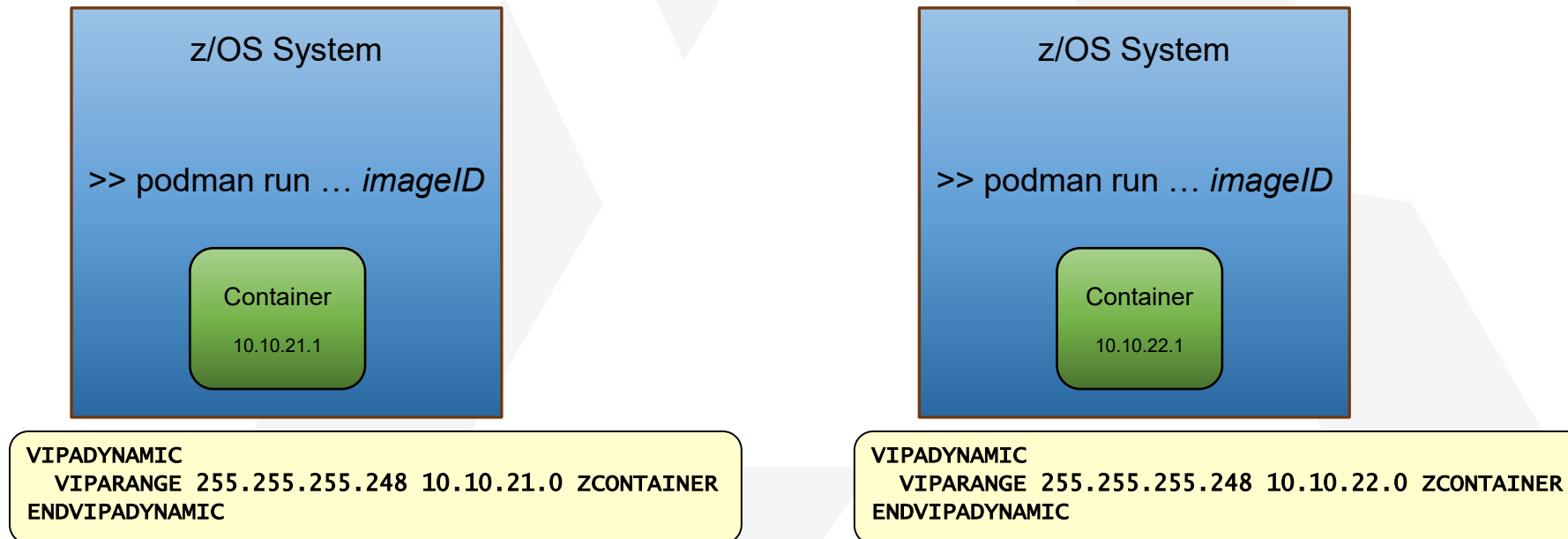
Configuring your network for Podman

Container Network Interface (CNI)

A **specification** to configure network interfaces in containers

- Concerned with adding, connecting and removing containers to/from networks via plugins
- Defines an interface for configuring the network, provisioning IP addresses, and maintaining connectivity with other containers and hosts
- <https://github.com/containernetworking/cni/blob/main/SPEC.md>

CNI plugins invoked by container runtimes such as Podman or cri-o (for Kubernetes)



CNI for z/OS Containers Platform

- Provides interface between container runtimes and z/OS Communications Server
- Configured with two plugins
 - zos-cni : APIs to assign VIPARANGE dynamic VIPAs to Pods or return them back to Communications Server
 - loopback : APIs to assign or return common loopback interface (127.0.0.1 or ::1)

```
{
  "cniVersion": "1.0.0",
  "name": "zos_hybrid_network",
  "disableCheck": false,
  "plugins": [
    {
      "type": "zos-cni",
      "ipv4": true,
      "ipv6": false
    },
    {
      "type": "loopback",
      "ipv4": true,
      "ipv6": false
    }
  ]
}
```

Network name for z/OS

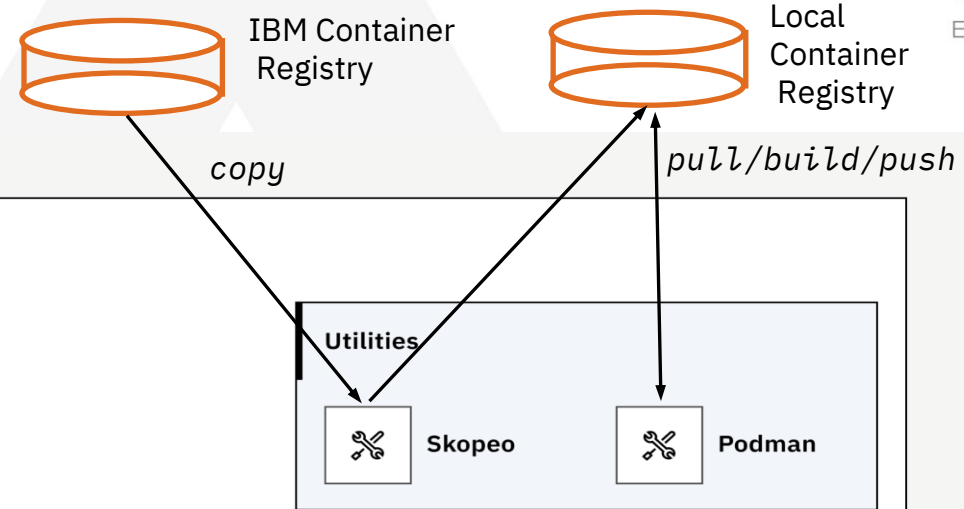
z/OS CNI plugins

Assign IPv4
VIPARANGE dynamic
VIPAs to Pods

- Cannot assign mix of IPv4 and IPv6 addresses

Building and running your first container image - demo

Pulling and building image



Pulling and building image...

```
>> podman images
REPOSITORY                TAG      IMAGE ID      CREATED      SIZE
localrepo.ibm.com/zoscp/zos latest 89123cdd6d1c 6 weeks ago 228 MB

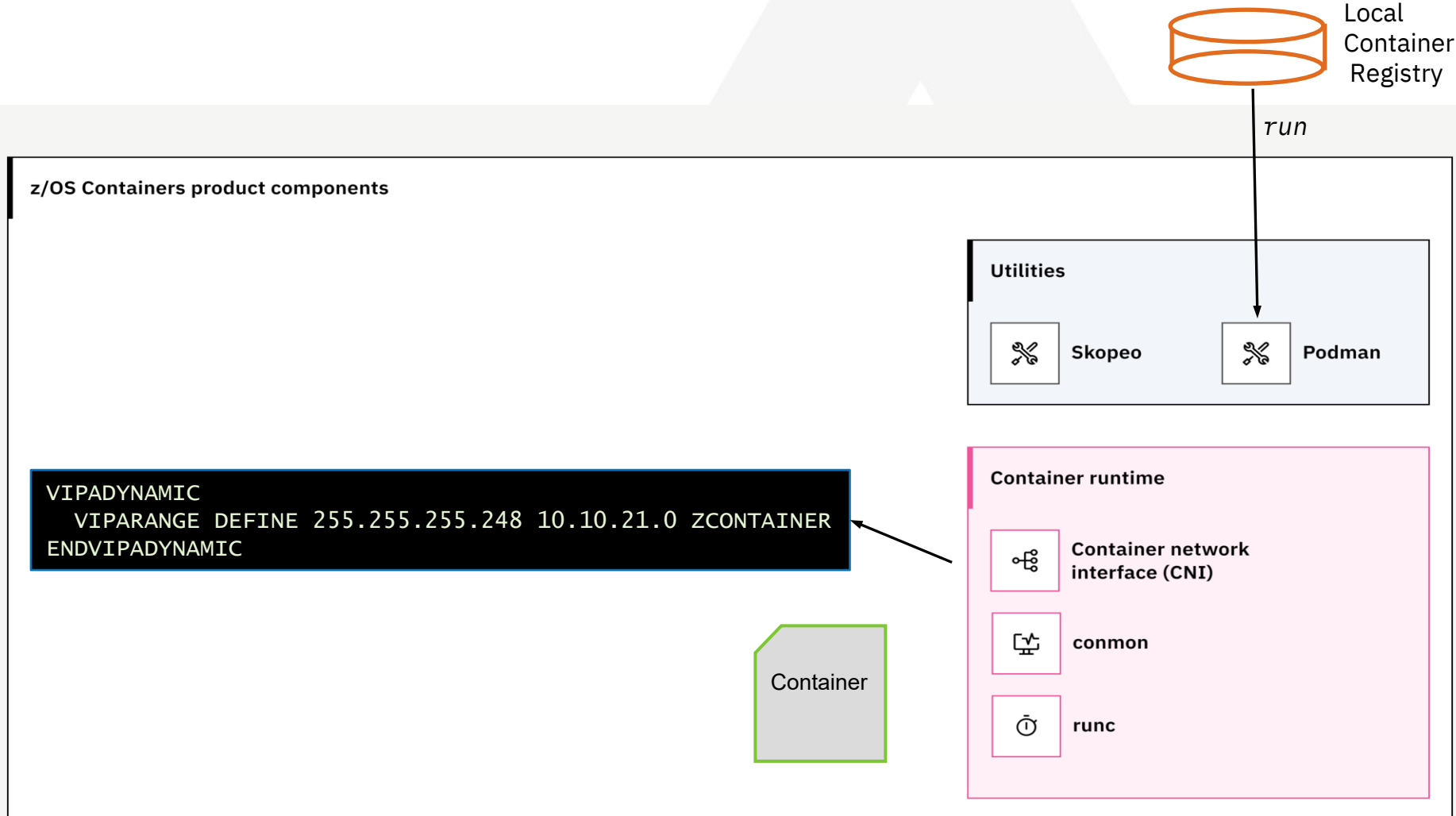
>> podman build . -t demo:1.0
STEP 1/5: FROM localrepo.ibm.com/zoscp/zos:latest
STEP 2/5: WORKDIR /
--> 8cc6fdc20f0
STEP 3/5: COPY demo ./
--> a2b6b7f874d
STEP 4/5: RUN chmod 555 demo
--> d2e32c9f213
STEP 5/5: CMD ["/demo"]
COMMIT demo:1.0
--> 7c42fe9cecb
Successfully tagged localhost/demo:1.0
7c42fe9cecb7e5c76a2726666da2a08be5a9e27ff34d7298d4e98377807bf01c

>> podman images
REPOSITORY                TAG      IMAGE ID      CREATED      SIZE
localhost/demo            1.0      7c42fe9cecb7 13 minutes ago 245 MB
localrepo.ibm.com/zoscp/zos latest 89123cdd6d1c 6 weeks ago 228 MB

>> podman tag localhost/demo:1.0 localrepo.ibm.com/demo:1.0

>> podman push localrepo.ibm.com/demo:1.0
Getting image source signatures
Copying blob 39c1a25583ad done
:
Copying config 7c42fe9cec done
writing manifest to image destination
Storing signatures
```

Running image



Running image...

```
>> netstat -F
Dynamic VIPA Information:
  VIPA Range:
    IpAddr/PrefixLen: 10.10.21.0/28
    Moveable: N/A
    Flg: P ← 'P' flag indicates ZCONTAINER range
```

```
>> podman images
REPOSITORY          TAG       IMAGE ID       CREATED        SIZE
localhost/demo      1.0       7c42fe9cecb7   17 minutes ago 245 MB
localrepo.ibm.com/demo 1.0       7c42fe9cecb7   17 minutes ago 245 MB
localrepo.ibm.com/zoscp/zos latest    89123cdd6d1c   6 weeks ago   228 MB
```

```
>> podman run -d -name demo localrepo.ibm.com/demo:1.0
9061c781139fbb124e57d8306747f8a41dad30309431d0a6f56e3fb80c2ed83f
```

```
>> podman ps
CONTAINER ID  IMAGE                COMMAND          CREATED        STATUS        PORTS  NAMES
9061c781139f localrepo.ibm.com/demo:1.0 ./demo          11 seconds ago Up 9 seconds ago  demo
```

```
>> podman inspect -f '{{.NetworkSettings.IPAddress}}' demo
10.10.21.1
```

```
>> curl http://10.10.21.1:8080

-> Hello from Pod 9061c781139f
```

```
>> netstat -c
User Id  Conn      State
-----  ----
BCZ-CNTR 00000146 Listen
Local Socket: 10.10.21.1..8080
Foreign Socket: 0.0.0.0..0
```

```
EZD1204I DYNAMIC VIPA 10.10.21.1 WAS CREATED USING IOCTL BY USER2 ON TCPIP
```

Dynamic VIPA activation message
issued to z/OS console

Multiple VIPARANGE statements for zOSCP

VIPARANGE ZCONTAINER statement

- Initial support allows for one VIPARANGE ZCONTAINER per address family
 - Obtaining a large range of contiguous IP addresses is challenging
 - Cannot use different subnets for users of both Podman and Kubernetes
 - Cannot define separate pools of IP addresses for different classes of users
- Allow for multiple VIPARANGE ZCONTAINER statements per address family
 - Supports both IPv4 and IPv6
 - Uses existing SAF keyword to provide separation
 - All ZCONTAINER ranges must either use SAF or not
 - If not, all addresses are considered in the same pool
 - New with 3.2 rolled back with PH63940 (2.5 / 3.1)

Sample Definitions

TCP Definitions

- VIPARANGE DEFINE 255.255.255.252 10.91.1.1 SAF PROFILE1 ZCONTAINER
- VIPARANGE DEFINE 255.255.255.252 10.92.2.2 SAF PROFILE2 ZCONTAINER
- VIPARANGE DEFINE 255.255.255.252 10.93.3.3 SAF PROFILE1 ZCONTAINER

SAF Definitions:

- RDEFINE SERVAUTH EZB.MODDVIPA.*sysname.tcpname*.PROFILE1 UACC(NONE)
- PERMIT EZB.MODDVIPA.*sysname.tcpname*.PROFILE1 CLASS(SERVAUTH) ID(USER1) ACCESS(READ)
- PERMIT EZB.MODDVIPA.*sysname.tcpname*.PROFILE2 CLASS(SERVAUTH) ID(USER2) ACCESS(READ)
- SETROPTS RACLIST(SERVAUTH) REFRESH

Demo

```
netstat -F
Dynamic VIPA Information:
```

```
IpAddr/PrefixLen: 10.91.1.1/30
```

```
Moveable: NA      SAFName: PROFILE1 Flg: P
```

```
IpAddr/PrefixLen: 10.92.2.2/30
```

```
Moveable: NA      SAFName: PROFILE2 Flg: P
```

```
IpAddr/PrefixLen: 10.93.3.3/30
```

```
Moveable: NA      SAFName: PROFILE1 Flg: P
```

```
>> podman run -d -name demo localrepo.ibm.com/demo:1.0
249f88236f105e604e184488f2f0eadd188c30036480348c518b9c5d6f66c683
```

Demo Container is started by
user1

```
>> podman run -d -name demo2 localrepo.ibm.com/demo:1.0
57778f71a287a106dbc8b37b117a1ae491a82f5d25030ed0feeb285b3585701d
```

Demo2 Container is Started by
user2

```
>> podman ps
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
249f88236f10	localrep.ibm.com/demo:1.0	./demo	20 seconds ago	Up 14 seconds ago		demo
57778f71a287	localrep.ibm.com/demo:1.0	./demo	11 seconds ago	Up 9 seconds ago		demo2

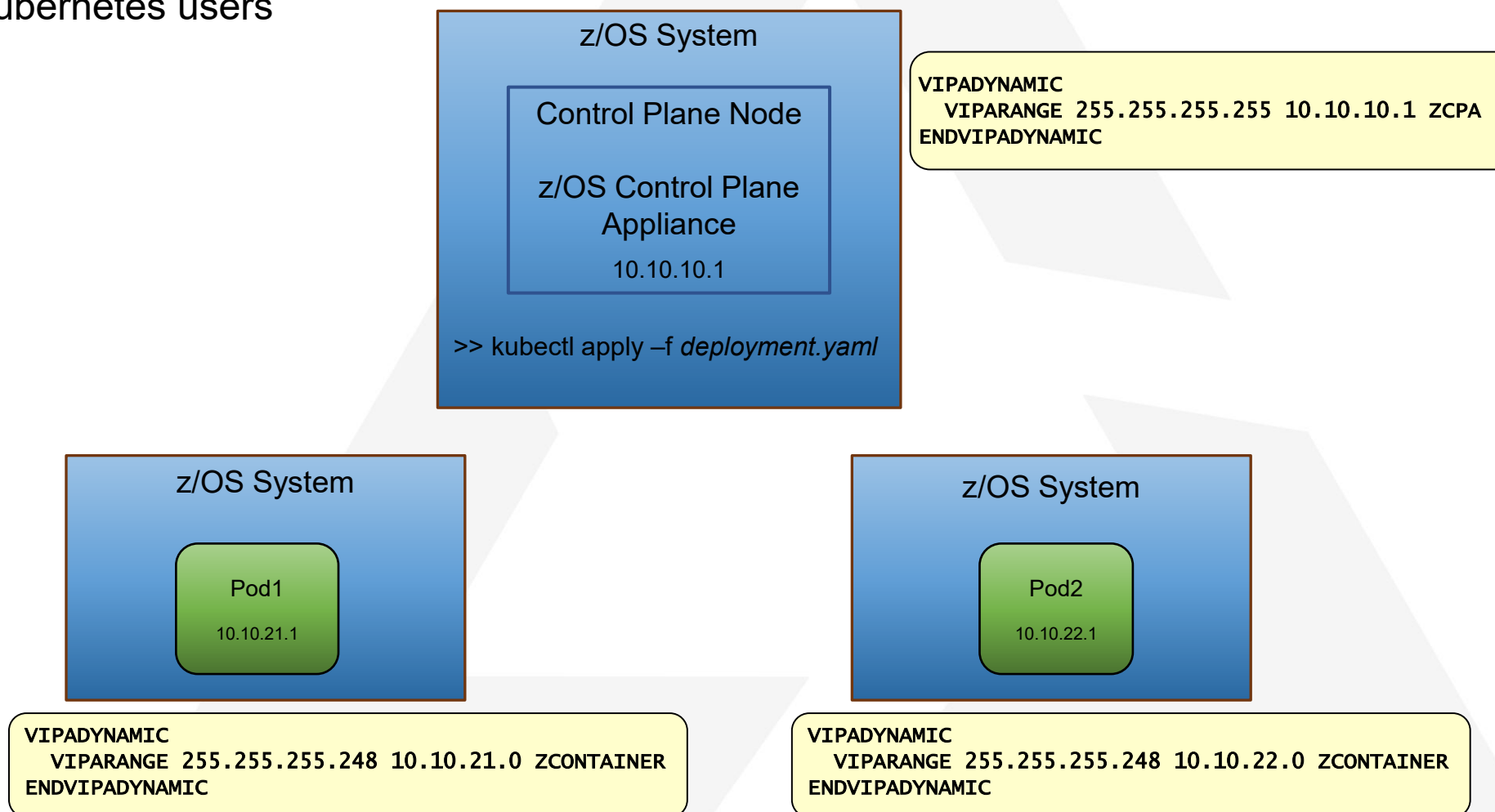
```
>> podman inspect -f '{{.NetworkSettings.IPAddress}}' demo
10.91.1.1
```

```
>> podman inspect -f '{{.NetworkSettings.IPAddress}}' demo2
10.92.2.1
```

Configuring your network for Kubernetes

z/OS Control Plane Appliance (zCPA) and Container Network Interface (CNI)

- Additional VIPARANGE statement for zCPAs
- By default VIPARANGE statement for container IP addresses shared between Podman and Kubernetes users



Provisioning a zCPA- network configuration

Workflows

Workflows > Provision a Control Plane Appliance (zCPA) > 1.2. Gather z/OS Control Plane Appliance instance properties

Settings | Help

Properties for Workflow Step 1.2. Gather z/OS Control Plane Appliance instance properties

General Details Dependencies Notes **Perform** Status Input Variables Feedback

✓ Input Variables

- ✓ z/OS Control Plane Appliance General Configuration
- ✓ z/OS Control Plane Appliance General Data Set Configuration
- ✓ z/OS Control Plane Appliance CPU and Memory Configuration
- ✚ **z/OS Control Plane Appliance Network Configuration**
- z/OS Control Plane Appliance Volume Serial Data Set Configuration
- z/OS Control Plane Appliance SMS Managed Data Set Configuration

Review Instructions

Input Variables - z/OS Control Plane Appliance Network Configuration

Enter the variable values for this input category.

* IPv4 Address: ⓘ - IPv4 address for the zCPA instance (as defined by a VIPARANGE ZCPA TCP/IP profile statement):

Sysplex Distributor IPv4 Address: ⓘ - IPv4 address of the Sysplex Distributor DVIPA for the zCPA instance:

TCP/IP Stack Name: ⓘ - z/OS TCP/IP stack name:

* MTU Size: ⓘ - The MTU to use for network communication with the zCPA instance:

1492

< Back Next > Save Finish Cancel

IPv4 address assigned in VIPARANGE ZCPA statement in the TCP/IP profile

Optionally, IPv4 address assigned as distributable dynamic VIPA in VIPADISTRIBUTE EXTARG statement in the TCP/IP profile

Initializing your Kubernetes cluster

Starting zCPA and initializing Kubernetes

```
S ZCPA,JOBNAME=ZCPA1,CONF='/global/zcpa/instances/ZCPA1/start.json'
```

```
GLZB025I zCX instance ZCPA1: Initialization has started. Code date 02/21/25.
```

```
EZD1204I DYNAMIC VIPA 10.10.10.1 WAS CREATED USING IOCTL BY ZCPA1 ON TCPIP
```

```
:
```

```
Workflows Performed:
```

```
Provision:          1.0.1      HCZ1110      2026/02/21 12:52
```

```
Reconfigure:        N/A        N/A          N/A
```

```
Upgrade :           N/A        N/A          N/A
```

```
Add Data Disks: :   N/A        N/A          N/A
```

```
GLZB027I zCX instance ZCPA1: Starting virtual CPUs.
```

```
GLZB027I zCX instance ZCPA1: IPLing guest.
```

```
GLZM012I zCX services for instance ZCPA1 are available.
```

VIPARANGE dynamic VIPA created and assigned to ZCPA1

When provisioning workflow was performed to create ZCPA1

ZCPA1 is active - control plane node can now be started

```
>> kubeadmz init --control-plane-appliance-address 10.10.10.1
```

```
BCZ2001I kubeadmz initialized the Kubernetes control plane on zCPA 10.10.10.1
```

```
>> kubeadmz getConfig --control-plane-appliance-address 10.10.10.1
```

```
BCZ2002I kubeadmz wrote zCPA Kubernetes config file /u/user2/.kube/config
```

Initialize control plane node - node name defaults to zCPA instance name

Extract Kubernetes configuration file and save in user's HOME directory - kubectl utility looks for configuration file here, by default

```
>> kubectl get nodes
```

NAME	STATUS	ROLES	AGE	VERSION
zcpa1	Ready	control-plane	13m	v1.34.3

Display all nodes in the Kubernetes cluster

Configuring and starting Kubernetes worker nodes

Additional z/OSMF workflow to configure each z/OS system as a worker node

- Beyond scope of this networking session

```
>> kubeadmz join 10.10.10.1
. . . . .
BCZ2218I kubeadmz initialized the z/OS worker node to the Kubernetes cluster
```

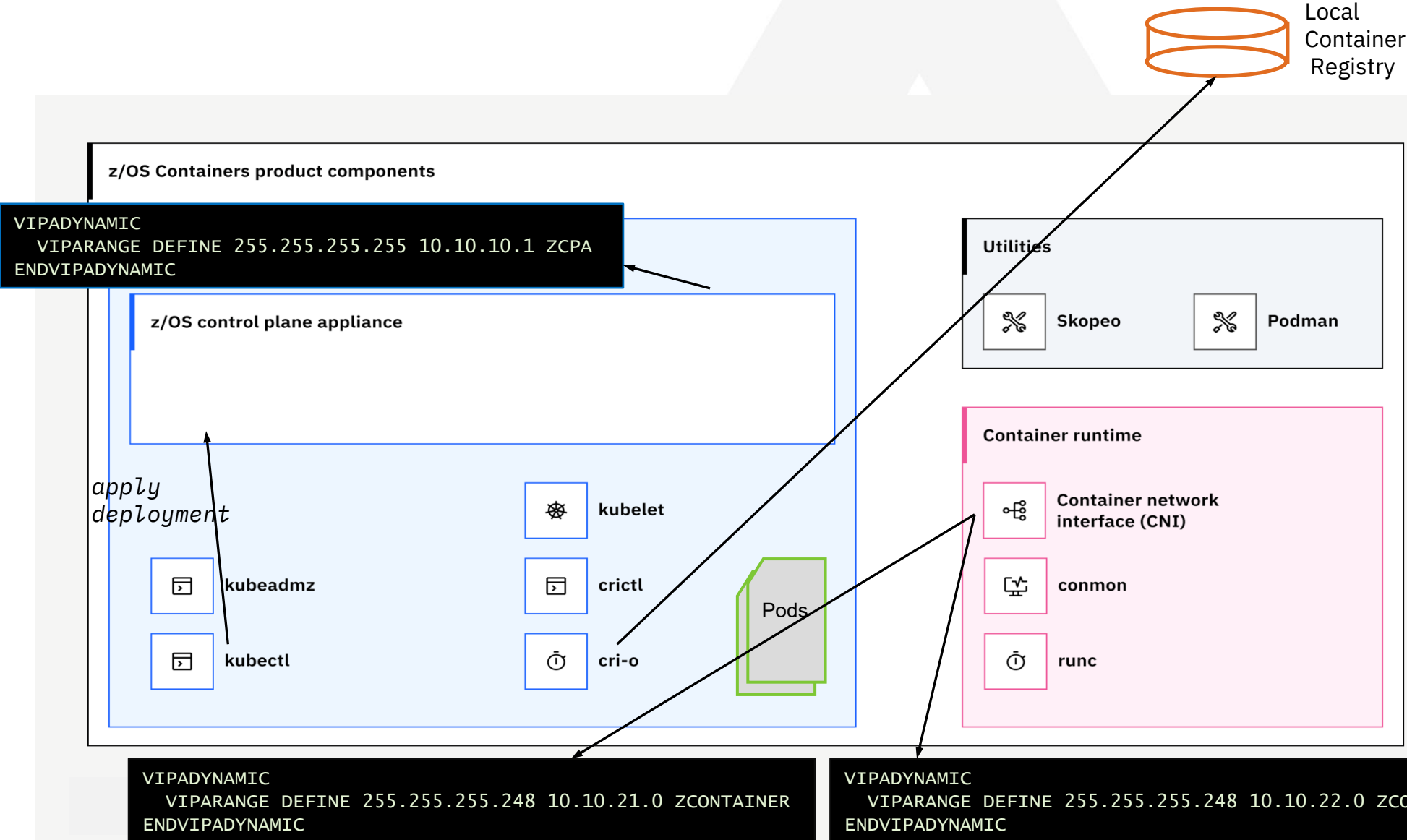
Join this z/OS system to the previously initialized control plane node

```
>> kubectl get nodes
NAME      STATUS    ROLES    AGE   VERSION
node1     Ready     <none>    72s   v1.31.2-zos.v1.1.0
node2     Ready     <none>    11m   v1.31.2-zos.v1.1.0
zcpa1     Ready     control-plane 38m   v1.34.3
```

Display all nodes in the Kubernetes cluster

Deploying your container image into a Kubernetes cluster

Deploying your image



Deploying image...

```
>> kubectl apply -f demo.yaml
deployment.apps/demo created
```

```
>> kubectl get pods -o wide
```

NAME	READY	STATUS	RESTARTS	AGE	IP
demo-67f5bddd46-gr1cw	1/1	Running	0	47s	10.10.21.1
node1					
demo-67f5bddd46-nvvzv	1/1	Running	0	47s	10.10.22.1
node2					

```
>> curl http://10.10.21.1:8080
```

```
-> Hello from Pod demo-67f5bddd46-nvvzv
```

```
>> curl http://10.10.22.1:8080
```

```
-> Hello from Pod demo-67f5bddd46-gr1cw
```

demo.yaml

```
apiVersion: apps/v1
kind: Deployment
metadata:
  labels:
    name: demo-deployment
    name: demo
spec:
  replicas: 2
  selector:
    matchLabels:
      name: demo-replica
  template:
    metadata:
      annotations:
        zoscp.ibm.com/networking: "hybrid"
        zoscp.ibm.com/userid: "ociuser"
    labels:
      name: demo-replica
  spec:
    nodeSelector:
      kubernetes.io/os: zos
    containers:
      - image: localrepo.ibm.com/demo:1.0
        imagePullPolicy: IfNotPresent
        name: demo
        securityContext:
          allowPrivilegeEscalation: false
```

EZD1204I DYNAMIC VIPA 10.10.21.1 WAS CREATED USING IOCTL BY CRIO4 ON TCPIP

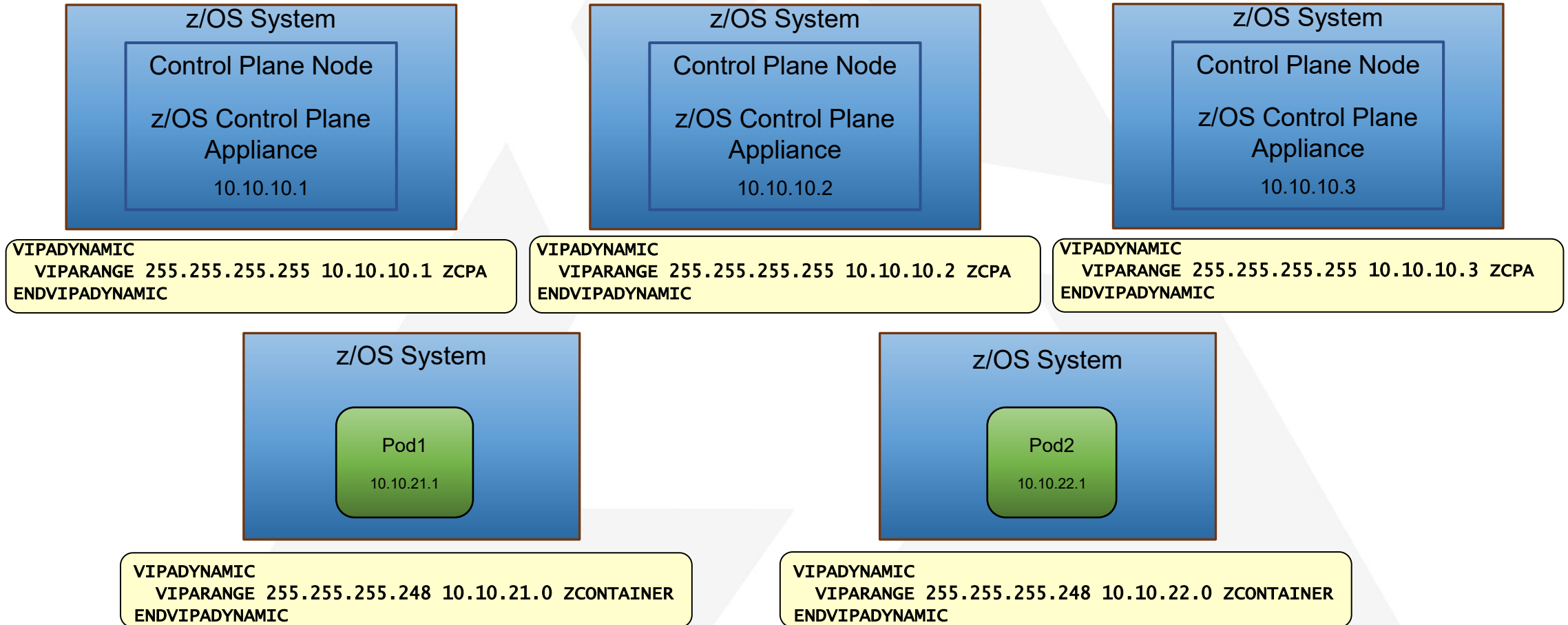
↑
Dynamic VIPA activation message
issued to z/OS console – user is
always CRIO

High Availability for your Kubernetes cluster

Multi-zCPA Kubernetes cluster

- Additional VIPARANGE statement for each zCPA
- New VIPADISTRIBUTE statement for external targets

```
VIPADISTRIBUTE DISTMETHOD ROUNDROBIN EXTTARG 10.10.20.1  
DESTIP 10.10.10.1 10.10.10.2 10.10.10.3
```



Multiple VIPARANGE statements for Kubernetes

TCP Definitions

- `VIPARANGE DEFINE 255.255.255.252 10.91.1.1 SAF PODS ZCONTAINER`
- `VIPARANGE DEFINE 255.255.255.252 10.92.2.2 SAF KUBE ZCONTAINER`
- `VIPARANGE DEFINE 255.255.255.252 10.93.3.3 SAF KUBE ZCONTAINER`

SAF Definitions:

- `RDEFINE SERVAUTH EZB.MODDVIPA.sysname.tcpname.KUBE UACC(NONE)`
- `PERMIT EZB.MODDVIPA.sysname.tcpname.KUBE CLASS(SERVAUTH) ID(OCIUSER*) ACCESS(READ)`
- `SETROPTS RACLIST(SERVAUTH) REFRESH`

Provisioning each zCPA- network configuration

Workflows

Workflows > Provision a Control Plane Appliance (zCPA) > 1.2. Gather z/OS Control Plane Appliance instance properties

Settings | Help

Properties for Workflow Step 1.2. Gather z/OS Control Plane Appliance instance properties

General Details Dependencies Notes **Perform** Status Input Variables Feedback

✓ Input Variables

- ✓ z/OS Control Plane Appliance General Configuration
- ✓ z/OS Control Plane Appliance General Data Set Configuration
- ✓ z/OS Control Plane Appliance CPU and Memory Configuration
- ➡ **z/OS Control Plane Appliance Network Configuration**
- z/OS Control Plane Appliance Volume Serial Data Set Configuration
- z/OS Control Plane Appliance SMS Managed Data Set Configuration

Review Instructions

Input Variables - z/OS Control Plane Appliance Network Configuration

Enter the variable values for this input category.

* IPv4 Address: ⓘ - IPv4 address for the zCPA instance (as defined by a VIPARANGE ZCPA TCP/IP profile statement):

Sysplex Distributor IPv4 Address: ⓘ - IPv4 address of the Sysplex Distributor DVIPA for the zCPA instance:

TCP/IP Stack Name: ⓘ - z/OS TCP/IP stack name:

* MTU Size: ⓘ - The MTU to use for network communication with the zCPA instance:

1492

< Back Next > Save Finish Cancel

IPv4 address assigned as distributable dynamic VIPA in VIPADISTRIBUTE EXTARG statement in the TCP/IP profile

Initializing Kubernetes and joining additional control plane nodes

```
>> kubeadmz init --control-plane-appliance-address 10.10.10.1 --control-plane-endpoint 10.10.20.1 --upload-certs
```

Initialize control plane node – specify Sysplex Distributor as load balancer

```
. . . . .  
BCZ2001I kubeadmz initialized the Kubernetes control plane on zCPA 10.10.10.1
```

```
>> kubeadmz join 10.10.20.1 --control-plane-appliance-address 10.10.10.2 --control-plane
```

Join other control plane nodes to the Kubernetes cluster – specify Sysplex Distributor as cluster to join

```
. . . . .  
BCZ2001I kubeadmz joined the Kubernetes control plane on zCPA 10.10.10.2
```

```
>> kubeadmz join 10.10.20.1 --control-plane-appliance-address 10.10.10.3 --control-plane
```

```
. . . . .  
BCZ2001I kubeadmz joined the Kubernetes control plane on zCPA 10.10.10.3
```

Extract Kubernetes configuration file and save in user's HOME directory – target any active control plane node

```
>> kubeadmz getConfig --control-plane-appliance-address 10.10.10.1
```

```
BCZ2002I kubeadmz wrote zCPA Kubernetes config file /u/user2/.kube/config
```

```
>> kubectl get nodes
```

Display all nodes in the Kubernetes cluster – kubectl request routed through Sysplex Distributor

NAME	STATUS	ROLES	AGE	VERSION
zcpa1	Ready	control-plane	13m	v1.34.3
zcpa2	Ready	control-plane	11m	v1.34.3
zcpa3	Ready	control-plane	8m	v1.34.3

Initializing Kubernetes and Joining Additional Control Plane Nodes

```
>> netstat -F
:
VIPA Distribute:
  Dest:      10.10.20.1..N/A
  DestXCF:   10.10.10.1
  DistMethod: Roundrobin
  SysPt:     N/A  TimAff: No   Flg:  EXT
  Dest:      10.10.20.1..N/A
  DestXCF:   10.10.10.2
  DistMethod: Roundrobin
  SysPt:     N/A  TimAff: No   Flg:  EXT
  Dest:      10.10.20.1..N/A
  DestXCF:   10.10.10.3
  DistMethod: Roundrobin
  SysPt:     N/A  TimAff: No   Flg:  EXT
```

EXTTARG distribution to group of zCPAs

Dynamically discover server applications started on control plane nodes (in this case, the Kubernetes apiserver)

```
>> netstat -O
Dynamic VIPA Destination Port Table for TCP/IP stacks:
Dest:      10.10.20.1..6443
DestXCF:   10.10.10.1
TotalConn: 00000000111  Rdy: 001  WLM: 01  TSR: 100
DistMethod: Roundrobin
Flg: Dynamic, EXT
Dest:      10.10.20.1..6443
DestXCF:   10.10.10.2
TotalConn: 00000000073  Rdy: 001  WLM: 01  TSR: 100
DistMethod: Roundrobin
Flg: Dynamic, EXT
Dest:      10.10.20.1..6443
DestXCF:   10.10.10.3
TotalConn: 00000000057  Rdy: 001  WLM: 01  TSR: 100
DistMethod: Roundrobin
Flg: Dynamic, EXT
```

Deploying NodePort service

```
>> kubectl apply -f service.yaml  
service/demo created
```

```
>> netstat -n  
Dynamic VIPA Destination Port Table for TCP/IP stacks:  
Dest:      10.10.20.1..6443  
Target:     10.10.10.1  
TotalConn: 00000000111 Rdy: 001 WLM: 01 TSR: 100  
DistMethod: Roundrobin  
Flg: Dynamic, EXT  
:  
Dest:      10.10.20.1..30089  
Target:     10.10.10.1  
TotalConn: 00000000003 Rdy: 001 WLM: 01 TSR: 100  
DistMethod: Roundrobin  
Flg: Dynamic, EXT  
Dest:      10.10.20.1..30089  
Target:     10.10.10.2  
TotalConn: 00000000002 Rdy: 001 WLM: 01 TSR: 100  
DistMethod: Roundrobin  
Flg: Dynamic, EXT  
Dest:      10.10.20.1..30089  
Target:     10.10.10.3  
TotalConn: 00000000002 Rdy: 001 WLM: 01 TSR: 100  
DistMethod: Roundrobin  
Flg: Dynamic, EXT
```

```
>> curl http://10.10.20.1:30089
```

```
-> Hello from Pod demo-67f5bddd46-nvvzv
```

```
>> curl http://10.10.20.1:30089
```

```
-> Hello from Pod demo-67f5bddd46-grlcw
```

Service yaml file

- Create NodePort service on control plane node for port 30089

Dynamically discover NodePort services started on control plane nodes (in this case, NodePort 30089)

Targeting NodePort using Sysplex Distributor

- Distributed to a control plane node within one of the targeted zCPAs
- Service round-robins requests between both Pods

Questions?

Experience more with IBM



Visit us at the IBM Booth #113

After a full day of technical sessions, take a break with us!

Connect with our experts, snap a photo with the z17 Plexi or the latest Telum II, and get an up-close look at our Spyre Accelerator.

Come back each day for fresh topics and demos at our expert stations.

Think 2026

Join 5000+ senior business and technology leaders who are seizing the AI revolution to unlock unprecedented growth and productivity at **Think 2026**.

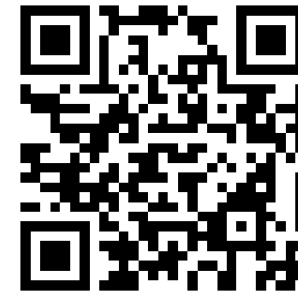
Find out more information using the QR code below.



IBM Digital Asset Haven

IBM Digital Asset Haven is the operational backbone for financial institutions and regulated enterprises entering the digital asset economy.

Find out more information using the QR code below.



Your feedback is important!

Submit a session evaluation for each session you attend:

www.share.org/evaluation



Thank you

Digital Badges & Online Courses

Start your **free IBM Z online learning** and earn IBM open badges!



Networking on z/OS - Foundations

Foundational understanding of networking on z/OS.

- **IBM Open Badge:**
<https://ibm.biz/zosnetworkingbadge>

- **Online course:**
<https://ibm.biz/zosnetworkingcourse>



z/OS TCP/IP Configuration with NCA

Use the IBM Configuration Assistant for z/OS Communications Server (NCA) to create and manage TCP/IP profiles.

- **IBM Open Badge:**
<http://ibm.biz/NCABadge>

- **Online course:**
<http://ibm.biz/NCATCPIPcourse>

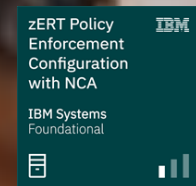


z/OS Network Security - Foundations

Knowledge and foundational understanding of z/OS network security.

- **IBM Open Badge:**
<http://ibm.biz/zosnetsecuritybadge>

- **Online course:**
<http://ibm.biz/zosnetsecuritycourse>

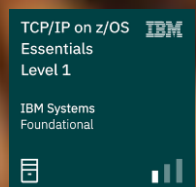


zERT Policy Enforcement Configuration with NCA

Configure zERT Policy Enforcement using the IBM Configuration Assistant for z/OS Communications Server (NCA)

- **IBM Open Badge:**
http://ibm.biz/NCA_zERTbadge

- **Online course:**
http://ibm.biz/NCA_zERTcourse



TCP/IP on z/OS Essentials - Level 1

General knowledge and understanding of TCP/IP on z/OS, including network layers, protocols at each layer, and the hardware that facilitates the transport of data.

- **IBM Open Badge:**
<http://ibm.biz/tcpipl1badge>

- **Online course:**
<https://ibm.biz/tcpipl1course>

Join Us on the IBM Community!

The z/OS Communications Server page on the IBM Community provides rich and up-to-date technical content including blogs, videos, and event information.

Join us at our new home:

<https://www.ibm.com/community/z/software/comm-server/>



Scan the QR code to visit the z/OS Communications Server home page on IBM Community.

z/OS Communications Server

A high-performance foundation for building and deploying networking applications on z/OS

Login or register

[Home](#)[Blog entries](#)[Discussions](#)[Events](#)[Videos](#)[Library](#)[Members](#)

Blog entries

[RSS](#) [Filter](#)

Digital Badge: IBM Shared Memory Communications Essentials

z/OS Communications Server by Erin ZHANG | Posted on 07/15/2022

Get your Digital Badge: IBM Shared Memory Communications Essentials today! Badge Description The badge earner has the knowledge and a good understanding of Shared Memory Communications (SMC). The individual can explain what SMC is, articulate its...

z/OS Comm Server

Demo Series: Configuring Shared Memory Communications (SMC) related items with Network Configuration Assistant

z/OS Communications Server by Xiao Xia Mao | Posted on 07/13/2022

This demo series shows how to use Network Configuration Assistant, which is a z/OSMF plugin, to configure Shared Memory Communications (SMC) related items. Configuring Shared Memory Communications (SMC) SMF Records (Available in V2R3 Network Configuration...

[IBM Z](#)[Software](#)[z/OS Comm Server](#)[z/OS](#)[z/OSMF](#)

ENS event calendar

z/OS Communications Server by Flora Gui | Posted on 07/07/2022

Welcome to ENS event calendar page. Being the one-stop page for virtual and off-line tech events of ENS offerings, this page provides you useful forecast and summary of diverse ENS related events. 2022 eventsz/OSMF Community Guild - IBM zERT Network...

[IBM Z Hardware](#)[IBM Z](#)[Software](#)[z/OS Comm Server](#)[z/OS](#)

Notices and Disclaimers



- © 2026 International Business Machines Corporation. No part of this document may be reproduced or transmitted in any form without written permission from IBM.
- **U.S. Government Users Restricted Rights — use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM.**
- Information in these presentations (including information relating to products that have not yet been announced by IBM) has been reviewed for accuracy as of the date of initial publication and could include unintentional technical or typographical errors. IBM shall have no responsibility to update this information. **This document is distributed “as is” without any warranty, either express or implied. In no event, shall IBM be liable for any damage arising from the use of this information, including but not limited to, loss of data, business interruption, loss of profit or loss of opportunity.** IBM products and services are warranted per the terms and conditions of the agreements under which they are provided.
- IBM products are manufactured from new parts or new and used parts. In some cases, a product may not be new and may have been previously installed. Regardless, our warranty terms apply.”
- **Any statements regarding IBM's future direction, intent or product plans are subject to change or withdrawal without notice.**
- Performance data contained herein was generally obtained in a controlled, isolated environments. Customer examples are presented as illustrations of how those
- customers have used IBM products and the results they may have achieved. Actual performance, cost, savings or other results in other operating environments may vary.
- References in this document to IBM products, programs, or services does not imply that IBM intends to make such products, programs or services available in all countries in which IBM operates or does business.
- Workshops, sessions and associated materials may have been prepared by independent session speakers, and do not necessarily reflect the views of IBM. All materials and discussions are provided for informational purposes only, and are neither intended to, nor shall constitute legal or other guidance or advice to any individual participant or their specific situation.
- It is the customer’s responsibility to insure its own compliance with legal requirements and to obtain advice of competent legal counsel as to the identification and interpretation of any relevant laws and regulatory requirements that may affect the customer’s business and any actions the customer may need to take to comply with such laws. IBM does not provide legal advice or represent or warrant that its services or products will ensure that the customer follows any law.

Notices and Disclaimers



- 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 about this publication 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. IBM does not warrant the quality of any third-party products, or the ability of any such third-party products to interoperate with IBM's products. **IBM expressly disclaims all warranties, expressed or implied, including but not limited to, the implied warranties of merchantability and fitness for a purpose.**
- The provision of the information contained herein is not intended to, and does not, grant any right or license under any IBM patents, copyrights, trademarks or other intellectual property right.
- IBM, the IBM logo, ibm.com and [names of other referenced IBM products and services used in the presentation] are trademarks of International Business Machines Corporation, registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the Web at "Copyright and trademark information" at: www.ibm.com/legal/copytrade.shtml

Trademarks



The following are trademarks of the International Business Machines Corporation in the United States and/or other countries.

IBM* IBM Logo*

* Registered trademarks of IBM Corporation

The following are trademarks or registered trademarks of other companies.

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 Office of Government Commerce, 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.

Docker and the Docker logo are trademarks or registered trademarks of Docker, Inc. in the United States and/or other countries. Docker, Inc. and other parties may also have trademark rights in other terms used herein.

* All other products may be trademarks or registered trademarks of their respective companies.

Notes:

Performance is in Internal Throughput Rate (ITR) ratio based on measurements and projections using standard IBM benchmarks in a controlled environment. The actual throughput that any user will experience will vary depending upon considerations such as the amount of multiprogramming in the user's job stream, the I/O configuration, the storage configuration, and the workload processed. Therefore, no assurance can be given that an individual user will achieve throughput improvements equivalent to the performance ratios stated here.

IBM hardware products are manufactured from new parts, or new and serviceable used parts. Regardless, our warranty terms apply.

All customer examples cited or described in this presentation are presented as illustrations of the manner in which some customers have used IBM products and the results they may have achieved. Actual environmental costs and performance characteristics will vary depending on individual customer configurations and conditions.

This publication was produced in the United States. IBM may not offer the products, services or features discussed in this document in other countries, and the information may be subject to change without notice. Consult your local IBM business contact for information on the product or services available in your area.

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

Information about non-IBM products is obtained from the manufacturers of those products or their published announcements. IBM has not tested those products and cannot confirm the 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.

Prices subject to change without notice. Contact your IBM representative or Business Partner for the most current pricing in your geography.