Compute Service from Scratch

Ikhsan Putra - Cloud Engineer

Bogor, 20 November 2020

boer technology

Introduction

Ikhsan Putra

Cloud Engineer

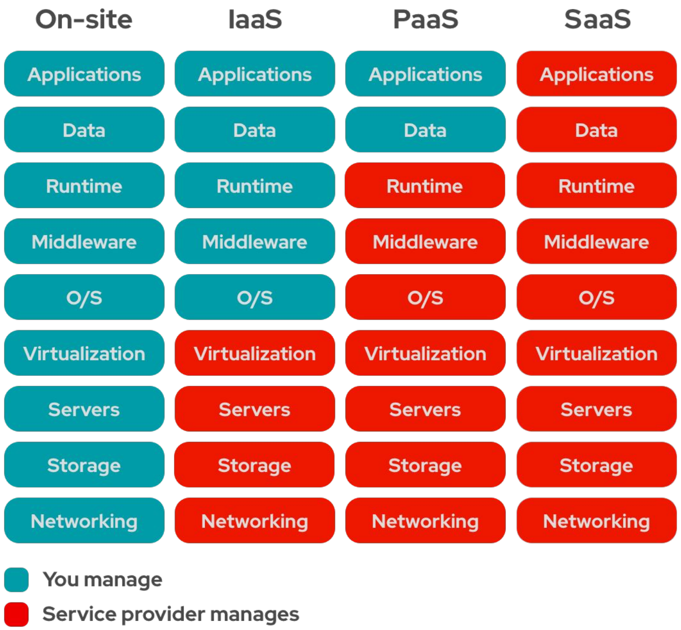
https://github.com/iputra/

https://ikhsanputra.com/

Objectives

- Understands compute on cloud (IaaS)
- Understands Qemu/KVM
- Understands overlay networking
- Start building compute services

Cloud Computing



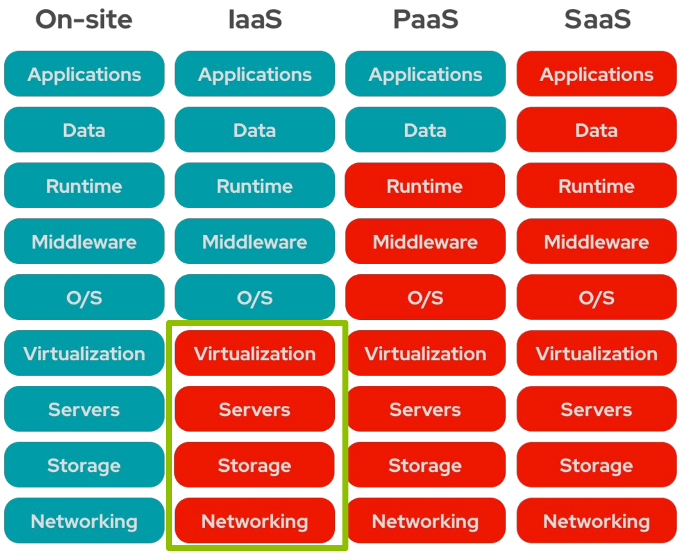
On-site	IaaS	PaaS	SaaS
🟦 Applications	🟦 Applications	🟦 Applications	🟥 Applications
🟦 Data	🟦 Data	🟦 Data	🟥 Data
🟦 Runtime	🟦 Runtime	🟥 Runtime	🟥 Runtime
🟦 Middleware	🟦 Middleware	🟥 Middleware	🟥 Middleware
🟦 O/S	🟦 O/S	🟥 O/S	🟥 O/S
🟦 Virtualization	🟥 Virtualization	🟥 Virtualization	🟥 Virtualization
🟦 Servers	🟥 Servers	🟥 Servers	🟥 Servers
🟦 Storage	🟥 Storage	🟥 Storage	🟥 Storage
🟦 Networking	🟥 Networking	🟥 Networking	🟥 Networking

🟦 You manage

🟥 Service provider manages

https://www.redhat.com/en/topics/cloud-computing/iaas-vs-paas-vs-saas

Cloud Computing



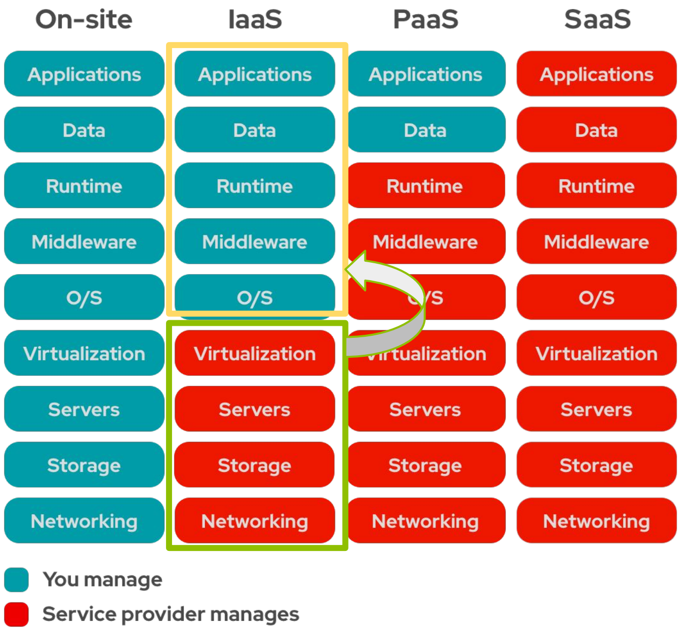
On-site	IaaS	PaaS	SaaS
🟦 Applications	🟦 Applications	🟦 Applications	🟥 Applications
🟦 Data	🟦 Data	🟦 Data	🟥 Data
🟦 Runtime	🟦 Runtime	🟥 Runtime	🟥 Runtime
🟦 Middleware	🟦 Middleware	🟥 Middleware	🟥 Middleware
🟦 O/S	🟦 O/S	🟥 O/S	🟥 O/S
🟦 Virtualization	🟥 Virtualization	🟥 Virtualization	🟥 Virtualization
🟦 Servers	🟥 Servers	🟥 Servers	🟥 Servers
🟦 Storage	🟥 Storage	🟥 Storage	🟥 Storage
🟦 Networking	🟥 Networking	🟥 Networking	🟥 Networking

🟦 You manage

🟥 Service provider manages

https://www.redhat.com/en/topics/cloud-computing/iaas-vs-paas-vs-saas

Cloud Computing



On-site	IaaS	PaaS	SaaS
🟦 Applications	🟦 Applications	🟦 Applications	🟥 Applications
🟦 Data	🟦 Data	🟦 Data	🟥 Data
🟦 Runtime	🟦 Runtime	🟥 Runtime	🟥 Runtime
🟦 Middleware	🟦 Middleware	🟥 Middleware	🟥 Middleware
🟦 O/S	🟦 O/S	🟥 O/S	🟥 O/S
🟦 Virtualization	🟥 Virtualization	🟥 Virtualization	🟥 Virtualization
🟦 Servers	🟥 Servers	🟥 Servers	🟥 Servers
🟦 Storage	🟥 Storage	🟥 Storage	🟥 Storage
🟦 Networking	🟥 Networking	🟥 Networking	🟥 Networking

🟦 You manage

🟥 Service provider manages

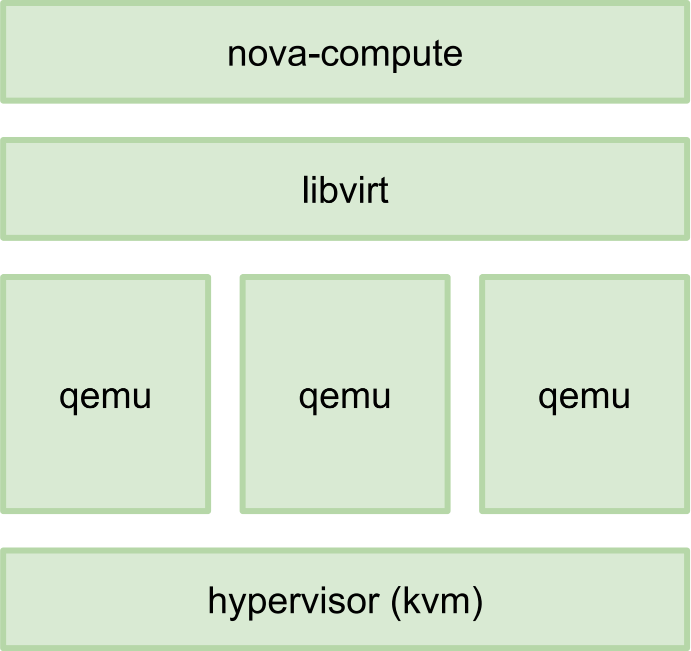
https://www.redhat.com/en/topics/cloud-computing/iaas-vs-paas-vs-saas

Compute Service

Nova is the OpenStack project that provides a way to provision compute instances (aka virtual servers). Nova supports creating virtual machines . . .

https://docs.openstack.org/nova/latest/

Compute Service



nova-compute

libvirt

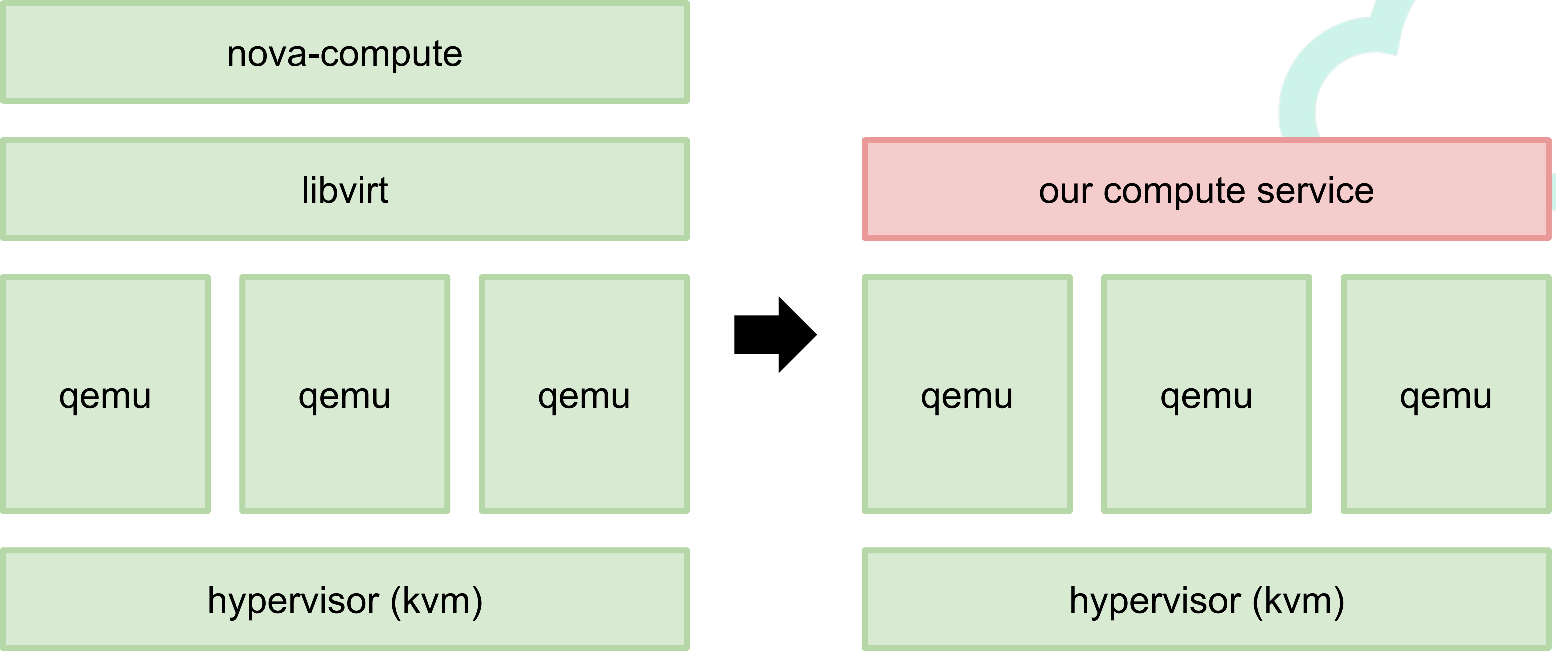
qemu

qemu

qemu

hypervisor (kvm)

Compute Service



Qemu/KVM

We present the internals of QEMU, a fast machine emulator using an original portable dynamic translator. It emulates several CPUs (x86, PowerPC, ARM and Sparc) on several hosts (x86, PowerPC, ARM, Sparc, Alpha and MIPS).

Bellard, F., 2005. QEMU, a fast and portable dynamic translator. In: USENIX 2005 Annual Technical Conference.

Qemu/KVM

The Kernel-based Virtual Machine, or kvm, is a new Linux subsystem which leverages these virtualization extensions to add a virtual machine monitor (or hypervisor) capability to Linux. Using kvm, one can create and run multiple virtual machines. These virtual machines appear as normal Linux processes and integrate seamlessly with the rest of the system.

Kivity, A., Lublin, U., Liguori, A., Kamay, Y. and Laor, D., 2007. kvm: the Linux virtual machine monitor. *Proceedings of the Linux Symposium*.

13'-0"
18'-6"
12'-4"

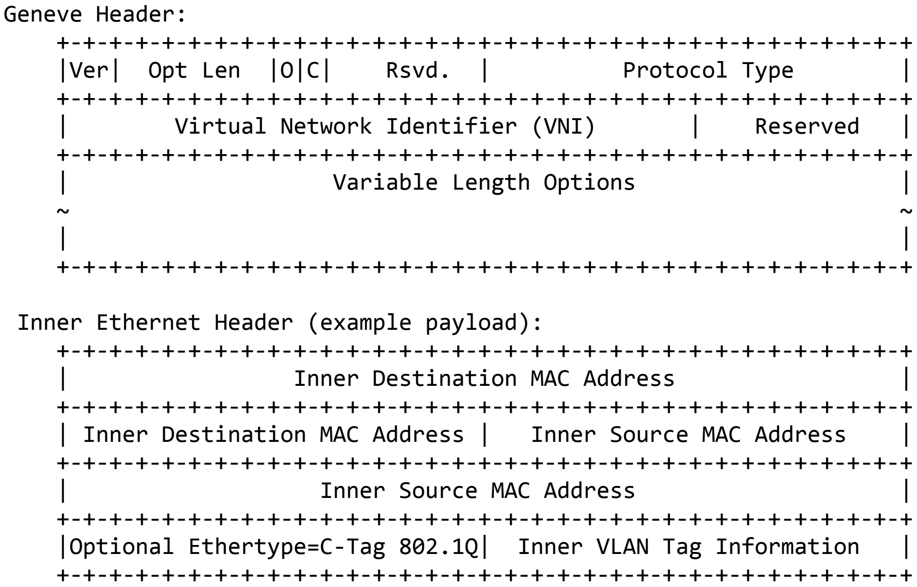
Overlay Network

It is intended for use in public or private data center environments, for deploying multi-tenant overlay networks over an existing IP underlay network.

https://tools.ietf.org/html/draft-ietf-nvo3-geneve-16

Kind of Overlay Network (Tunneling Network) :
GENEVE, VXLAN, NVGRE, STT, ...

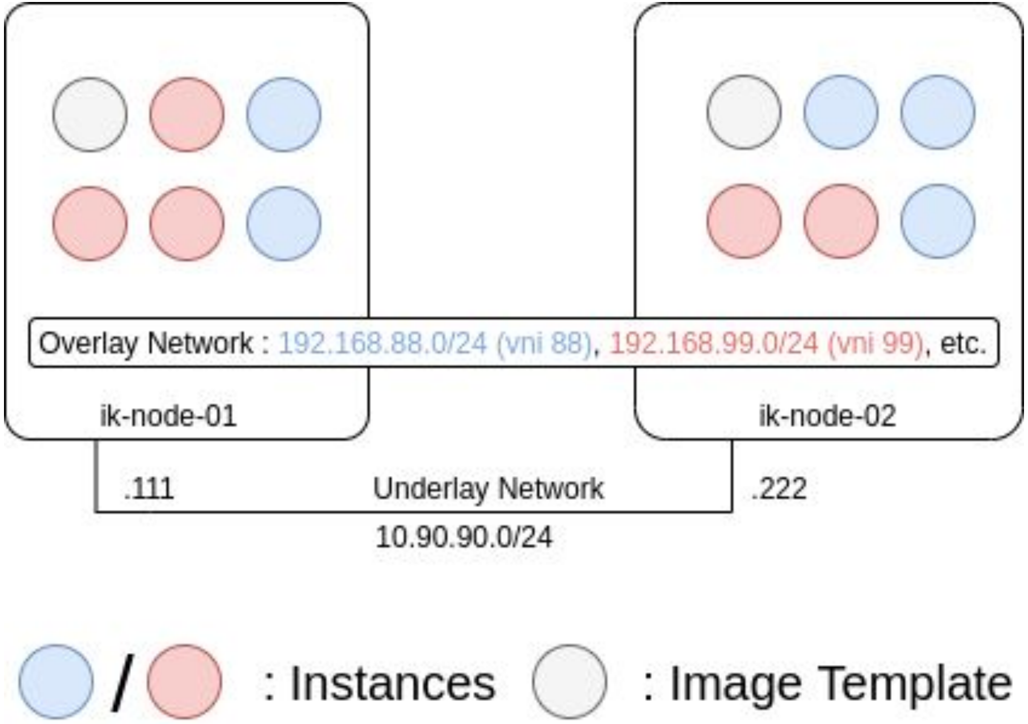
Overlay Network



```
Geneve Header:
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |Ver|  Opt Len  |O|C|    Rsvd.  |          Protocol Type        |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |        Virtual Network Identifier (VNI)       |    Reserved   |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |                    Variable Length Options                    |
    ~                                                               ~
    |                                                               |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

 Inner Ethernet Header (example payload):
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |                 Inner Destination MAC Address                 |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    | Inner Destination MAC Address |   Inner Source MAC Address    |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |                   Inner Source MAC Address                    |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |Optional Ethertype=C-Tag 802.1Q|  Inner VLAN Tag Information   |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

https://tools.ietf.org/html/draft-ietf-nvo3-geneve-16

Overlay Network



Overlay Network : 192.168.88.0/24 (vni 88), 192.168.99.0/24 (vni 99), etc.

ik-node-01

ik-node-02

.111

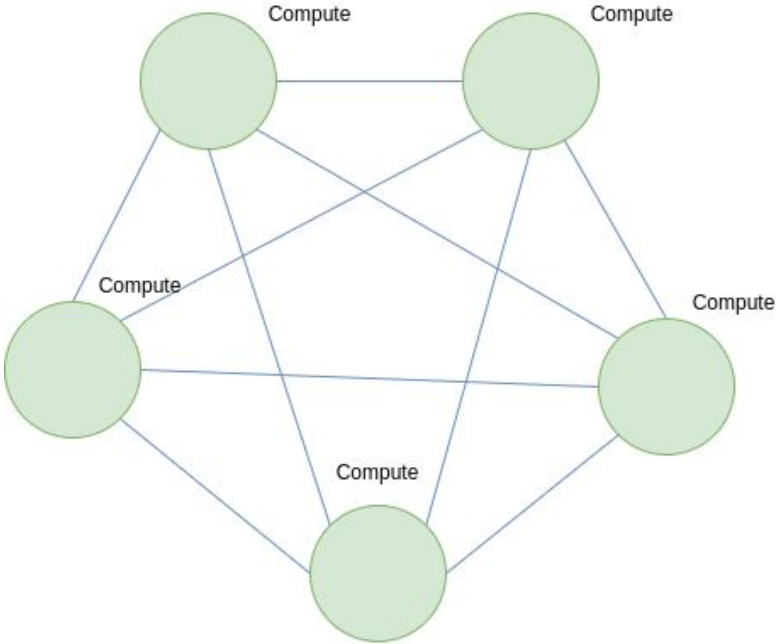
Underlay Network

.222

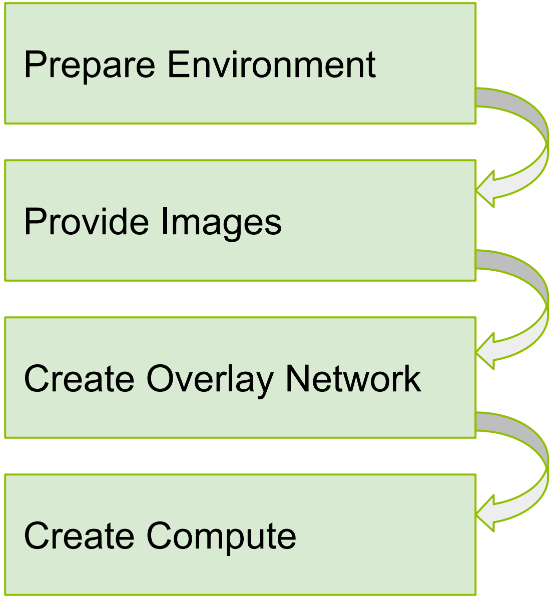
10.90.90.0/24

🔵 / 🔴 : Instances ⚪ : Image Template

Overlay Network



Start Building



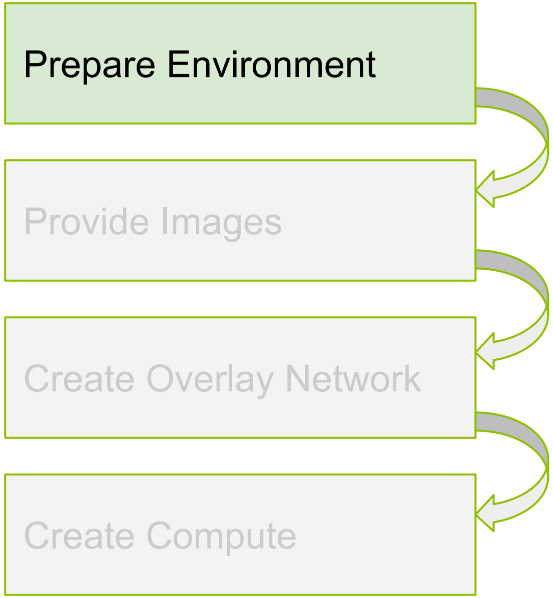
Prepare Environment

Provide Images

Create Overlay Network

Create Compute

Start Building

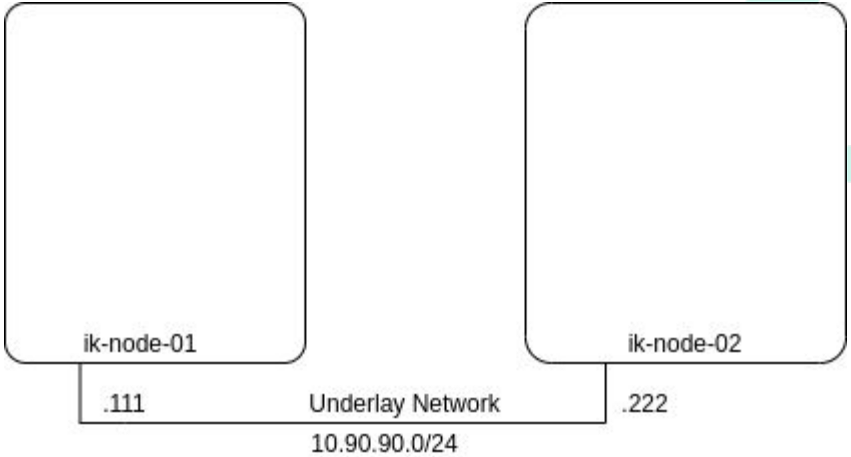


Prepare Environment

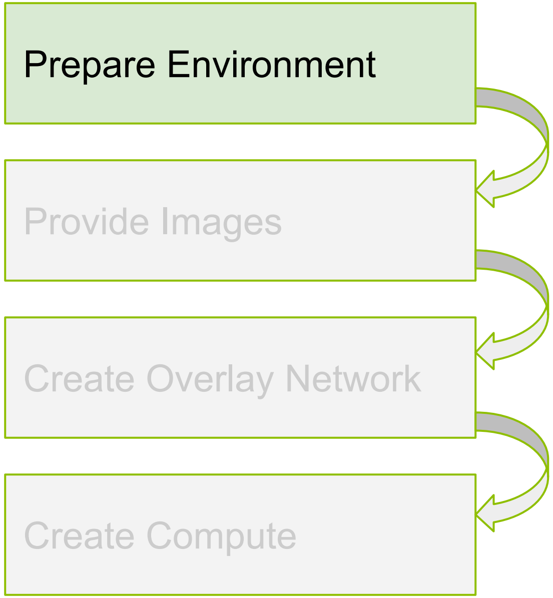
Provide Images

Create Overlay Network

Create Compute



Start Building


Prepare Environment

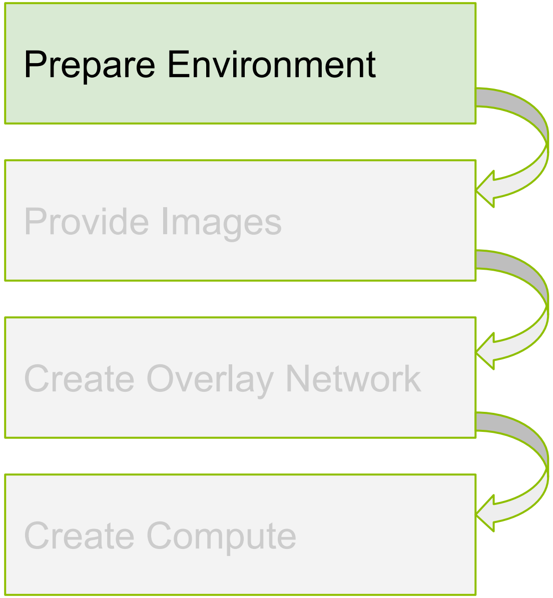
Provide Images

Create Overlay Network

Create Compute

```
compute-service/
. . /lib/
. . / . . /image/
. . / . . / . . /bionic-server-cloudimg-amd64.img

. . / . . /volumes/
. . / . . / . . /compute-af95e879-seed.qcow2
. . / . . / . . /compute-af95e879.img

. . /run/
. . / . . /instances/
. . / . . / . . /compute-af95e879/
. . / . . / . . / . . /cloud_init.cfg
. . / . . / . . / . . /compute-af95e879.pid
. . / . . / . . / . . /net_1.cfg
```

Start Building



Prepare Environment

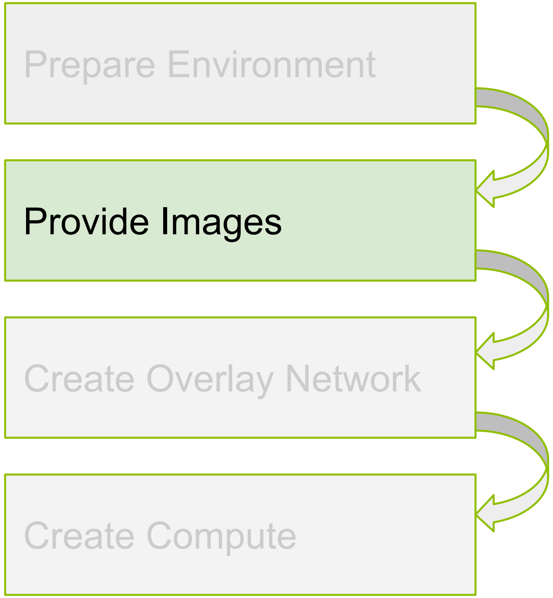
Provide Images

Create Overlay Network

Create Compute

```
# ik-node-01 and ik-node-02
# Install package
apt update; apt upgrade -y
apt install cloud-image-utils qemu-system-x86

# Create directory
mkdir compute-service; cd compute-service
mkdir -p lib/images lib/volumes
mkdir -p run/instances

# Set environment variable
export compute_service_root=/root/compute-service
export images_path=${compute_service_root}/lib/images
export volumes_path=${compute_service_root}/lib/volumes
export instances_path=${compute_service_root}/run/instances
```

Start Building



Prepare Environment

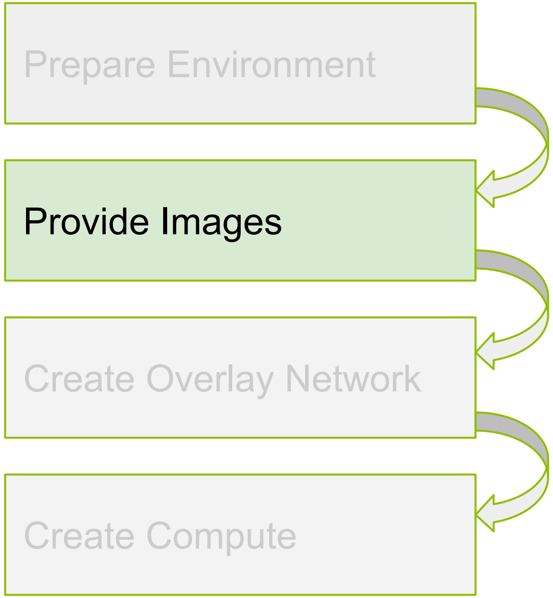
Provide Images

Create Overlay Network

Create Compute

```
# ik-node-01 and ik-node-02
# Download image
cd ${images_path}
wget
https://cloud-images.ubuntu.com/focal/current/focal-server-cloudimg-amd64.img

wget
https://cloud-images.ubuntu.com/bionic/current/bionic-server-cloudimg-amd64.img

wget
https://download.cirros-cloud.net/0.5.1/cirros-0.5.1-x86_64-disk.img

cd ${compute_service_root}
```

Start Building

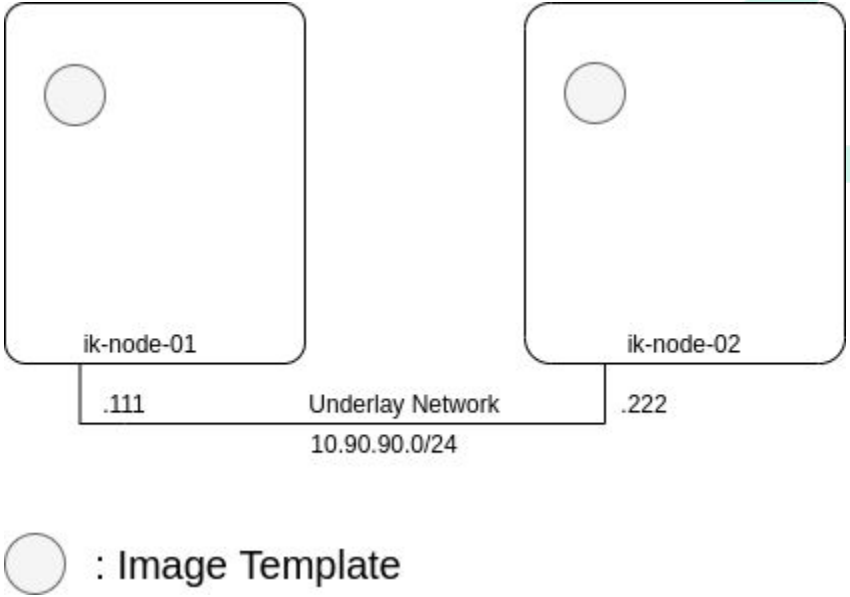


Prepare Environment

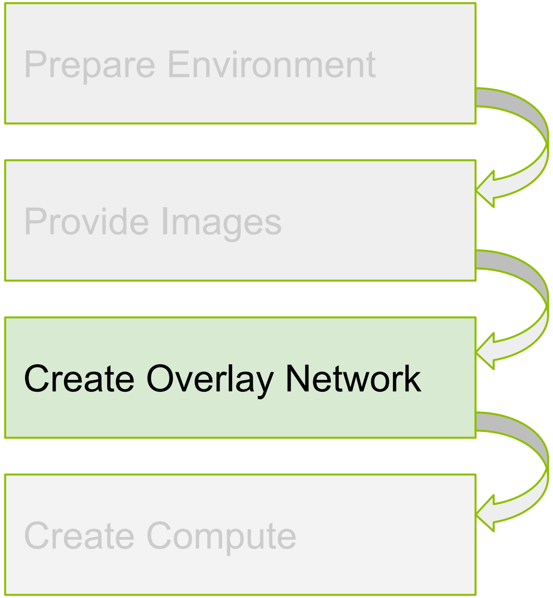
Provide Images

Create Overlay Network

Create Compute

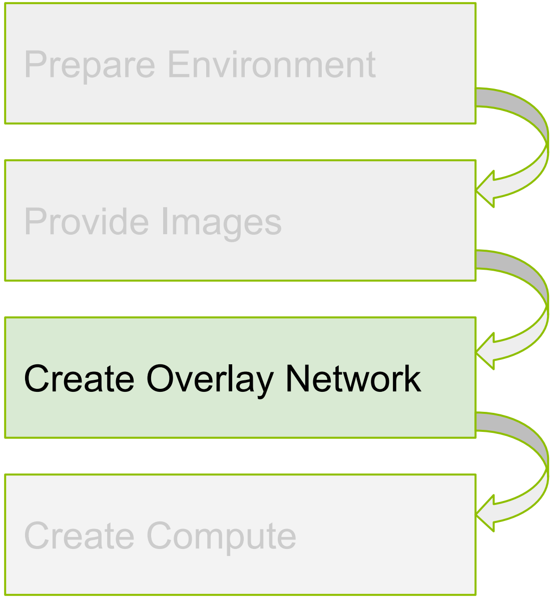


Start Building



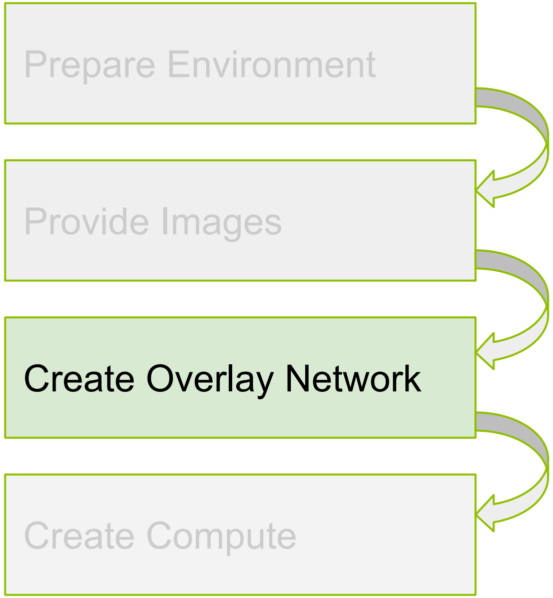
```
# ik-node-01
remote_host=10.90.90.222
vni=88
net_uuid=$(uuidgen -t | cut -d "-" -f1)

ip link add dev gnv-${net_uuid} type geneve remote ${remote_host} vni
${vni}
ip link set gnv-${net_uuid} up

ip link add br-gnv-${net_uuid} type bridge
ip link set dev br-gnv-${net_uuid} up
ip link set gnv-${net_uuid} master br-gnv-${net_uuid}

ip addr add 192.168.88.1/24 dev br-gnv-${net_uuid}
```

Start Building



```
# ik-node-01
# For instance get internet access
sysctl -w net.ipv4.ip_forward=1
iptables -t nat -A POSTROUTING -o ens3 -j MASQUERADE
iptables -A FORWARD -i br-gnv-${net_uuid} -o ens3 -j ACCEPT
```

Start Building



Prepare Environment

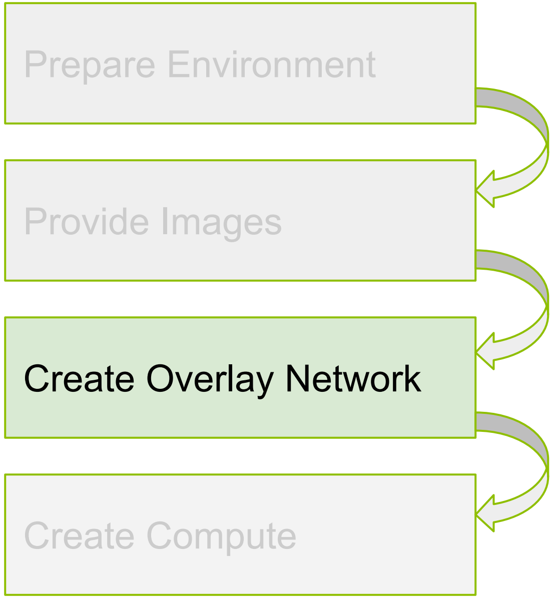
Provide Images

Create Overlay Network

Create Compute

```
# ik-node-01 configure to ik-node-02
remote_host=10.90.90.111
vni=88

ssh 10.90.90.222 /bin/bash <<EOF
  ip link add dev gnv-${net_uuid} type geneve \
    remote ${remote_host} vni ${vni}
  ip link set gnv-${net_uuid} up

  ip link add br-gnv-${net_uuid} type bridge
  ip link set dev br-gnv-${net_uuid} up
  ip link set gnv-${net_uuid} master br-gnv-${net_uuid}
EOF
```

Start Building

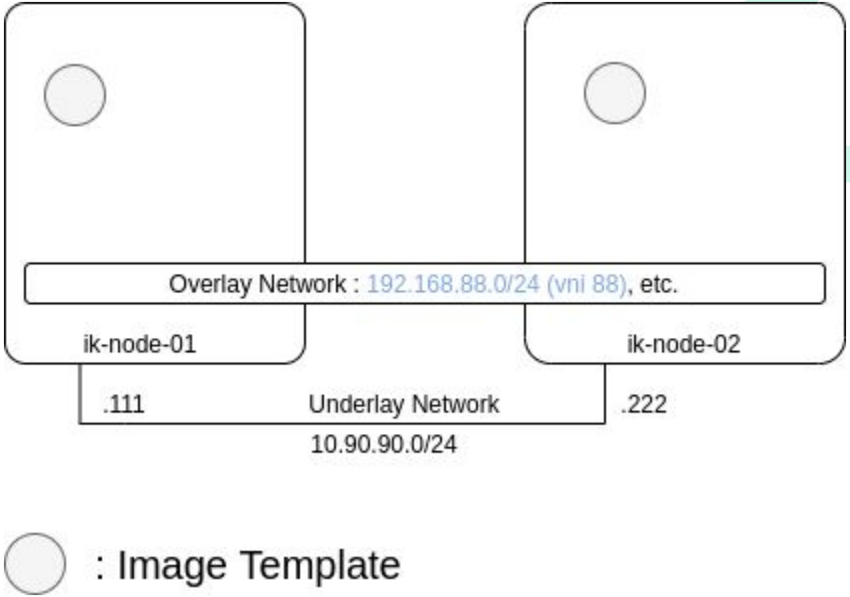


Prepare Environment

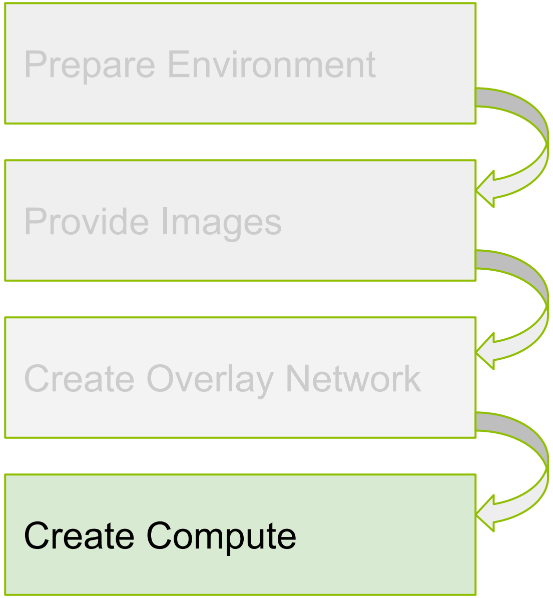
Provide Images

Create Overlay Network

Create Compute



Start Building



Prepare Environment

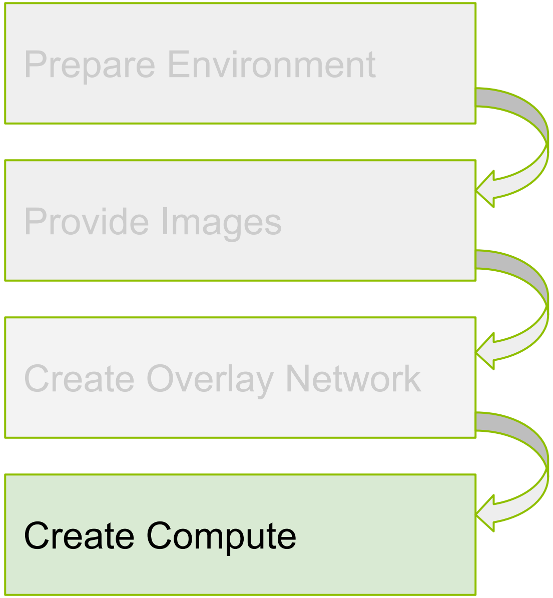
Provide Images

Create Overlay Network

Create Compute

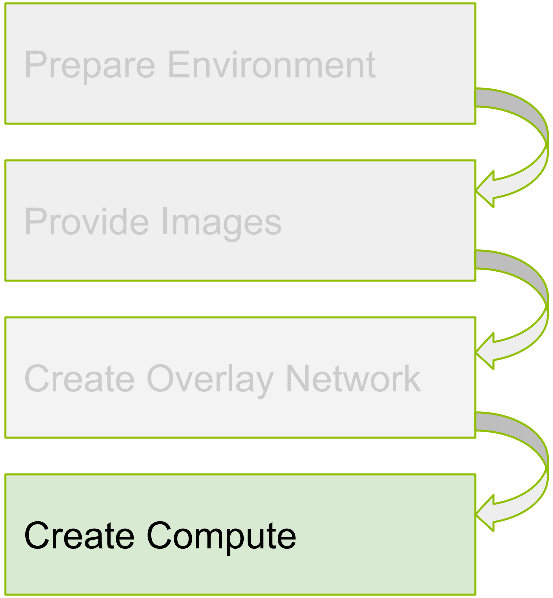
```
# ik-node-01
compute_uuid=$(uuidgen | cut -d '-' -f1)
pubkey=$(cat ~/.ssh/id_rsa.pub)
vnc_port=":1"
ip_addr="192.168.88.201"
gateway="192.168.88.1"
cpu=2
memory=4096
net_uuid=e106dc66
disk_size=10G
```

Start Building



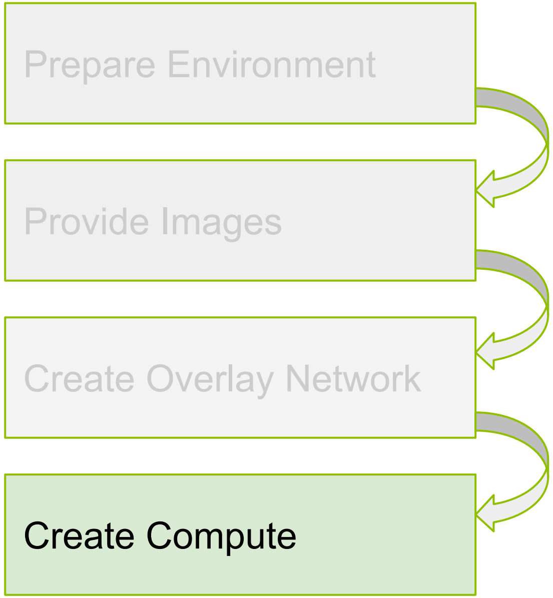
```
# ik-node-01
mkdir ${instances_path}/compute-${compute_uuid}
cat > ${instances_path}/compute-${compute_uuid}/net_1.cfg <<EOF
version: 2
ethernets:
  ens3:
     dhcp4: false
     addresses: [ ${ip_addr}/24 ]
     gateway4: ${gateway}
     nameservers:
       addresses: [ ${gateway},8.8.8.8 ]
EOF
```

Start Building



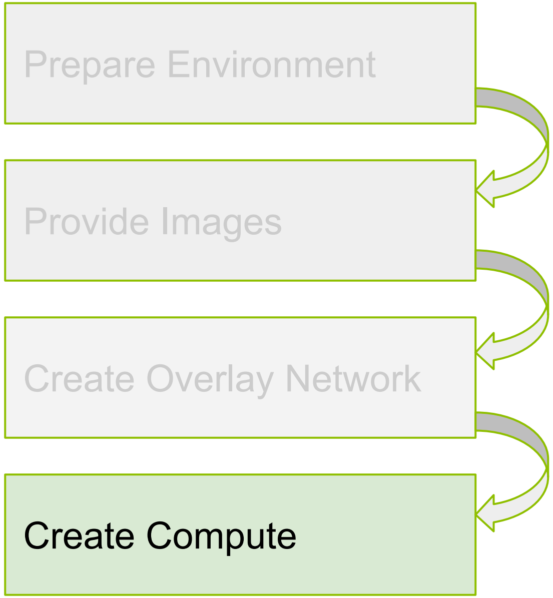
```
# ik-node-01
cat > ${instances_path}/compute-${compute_uuid}/cloud_init.cfg <<EOF
#cloud-config
hostname: ${compute_uuid}
users:
  - name: ubuntu
    sudo: ALL=(ALL) NOPASSWD:ALL
    groups: users, admin
    home: /home/ubuntu
    shell: /bin/bash
    lock_passwd: false
    ssh-authorized-keys:
      - ${pubkey}
ssh_pwauth: True
chpasswd:
  list: |
    ubuntu:ubuntu
  expire: False
EOF
```

Start Building



```
# ik-node-01
cloud-localds -v \
--network-config=${instances_path}/compute-${compute_uuid}/net_1.cfg \
${volumes_path}/compute-${compute_uuid}-seed.qcow2 \
${instances_path}/compute-${compute_uuid}/cloud_init.cfg

cp ${images_path}/bionic-server-cloudimg-amd64.img \
${volumes_path}/compute-${compute_uuid}.img

qemu-img resize ${volumes_path}/compute-${compute_uuid}.img ${disk_size}
```

Start Building



Prepare Environment

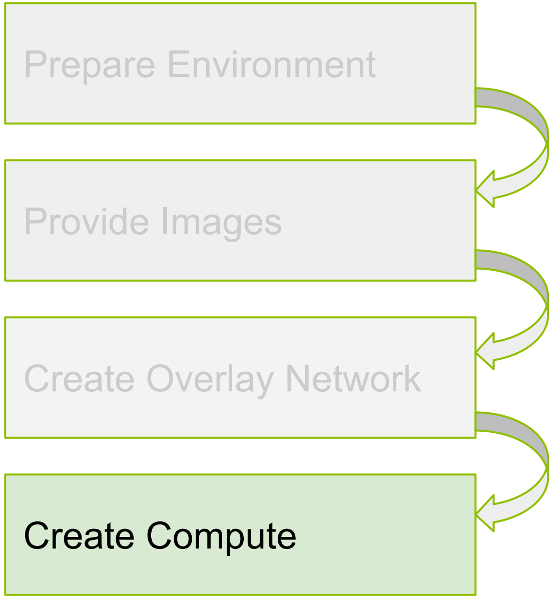
Provide Images

Create Overlay Network

Create Compute

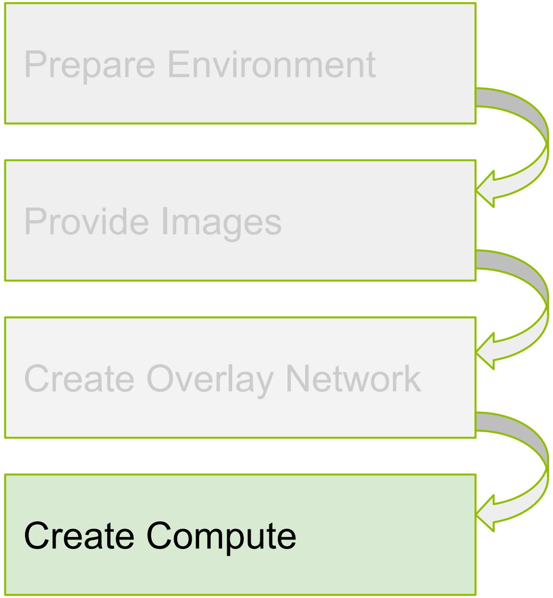
```
# ik-node-01
ip tuntap add dev vnet-${compute_uuid} mode tap
ip link set dev vnet-${compute_uuid} up
ip link set vnet-${compute_uuid} master br-gnv-${net_uuid}
```

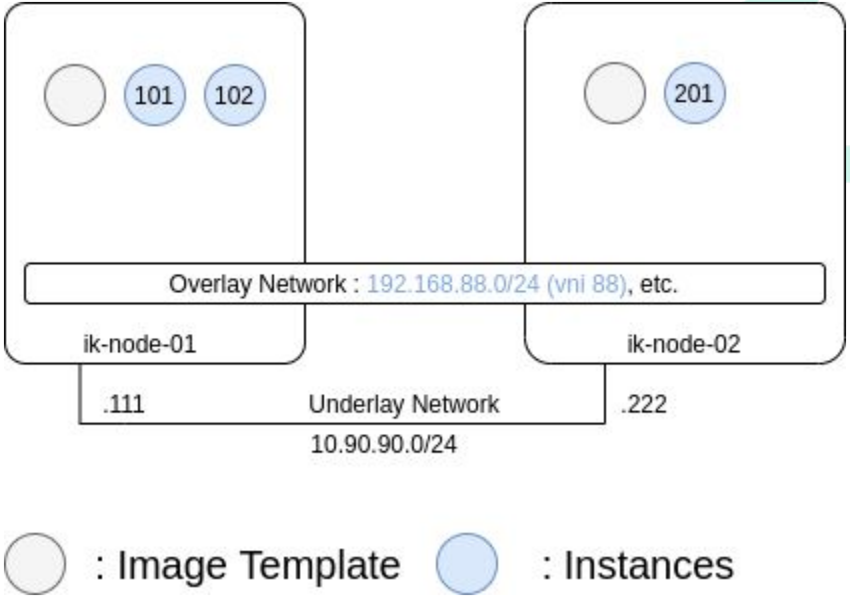
Start Building



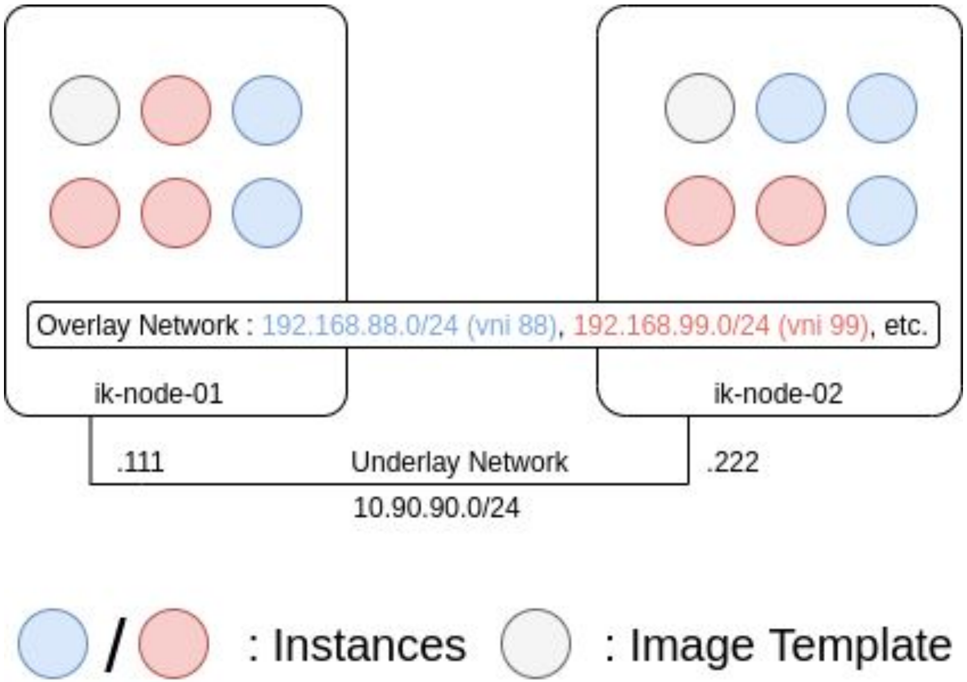
```
# ik-node-01
mac_addr=$(echo -n 02; od -t x1 -An -N 5 /dev/urandom | tr ' ' ':')
qemu-system-x86_64 -smp ${cpu} -m ${memory} \
  -drive file=${volumes_path}/compute-${compute_uuid}.img,format=qcow2,if=virtio \
  -drive file=${volumes_path}/compute-${compute_uuid}-seed.qcow2,format=raw,if=virtio \
  -boot order=c,menu=off \
  -enable-kvm \
  -cpu host \
  -device virtio-net-pci,netdev=network0,mac=${mac_addr} \
  -netdev tap,id=network0,ifname=vnet-${compute_uuid},script=no,downscript=no \
  -daemonize -vnc ${vnc_port} &

echo $! > ${instances_path}/compute-${compute_uuid}/compute-${compute_uuid}.pid
```

Start Building





Quizzz


Overlay Network : 192.168.88.0/24 (vni 88), 192.168.99.0/24 (vni 99), etc.
ik-node-01
ik-node-02
.111
Underlay Network
.222
10.90.90.0/24
/ : Instances
: Image Template

Extra Miles

Qemu : https://qemu.weilnetz.de/doc/2.7/qemu-doc.pdf

GENEVE : https://lwn.net/Articles/639265/

Tunnel Interfaces : https://developers.redhat.com/blog/2019/05/17/an-introduction-to-linux-virtual-interfaces-tunnels/

iproute2 : https://manpages.debian.org/testing/iproute2/ip-link.8.en.html

Thank You.

Question ?

Bogor, 20 November 2020

boer technology

The End.