

Práctico FCEFyN Redes de computadoras

Trabajo Práctico 6

Docentes: Matías R. Cuenca del Rey

Mail: mcuenca@unc.edu.ar

Natasha Tomattis

Mail: natitomattis@gmail.com

Colaborador externo: Matthew Aguerreberry

Ayudantes alumnos: Matías Kleiner, Elisabeth Leonhardt, Agustín Montero, Sergio Sulca

Redes de computadoras

Facultad de Ciencias Exactas, Físicas y Naturales

Universidad Nacional de Córdoba

Práctico 6: Aplicación.

Presentación de consignas.

Ejercicio 1: Ruteo Internet. Ruteo interno. Configuración de aplicaciones. Despliegue de aplicaciones.

Recomendaciones

- Lea con cuidado las consignas.
- Tenga certeza de los comandos que ejecuta.
- Para contenerización utilizar Docker CE.
- Para orquestación utilizar Kubernetes (K8s)

Esquema

- Se realizará bajo IPv4. No se usará IPv6.
- Se usarán máquinas virtuales en VirtualBox. La configuración de red debe ser Bridge (Adaptador puente).
- Se usarán containers dentro de una máquina virtual para el router y el servidor DNS.
- Se usará kubernetes dentro de otra máquina virtual para los distintos servicios.
- Todas las acciones sobre los servidores deben ser llevadas a cabo con el comando SSH.

Diagrama

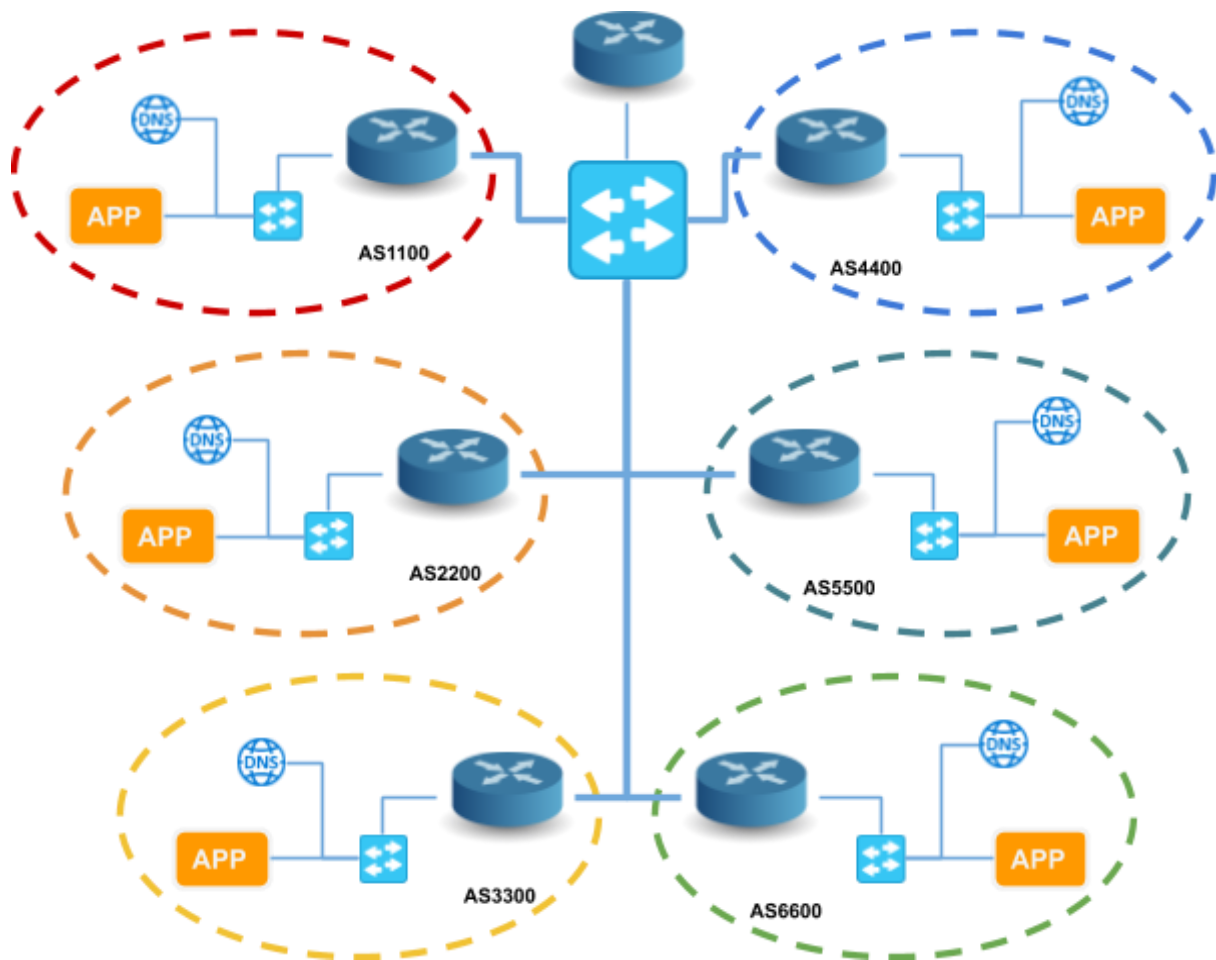


Tabla de asignación de direcciones IPv4

Completar la tabla de asignación de direcciones IP basado en los datos provistos por grupo.

Sistema Autónomo	Red asignada	Cantidad de hosts	Redes

Links de ayuda

Instalación de Docker CE: <https://docs.docker.com/install/linux/docker-ce/ubuntu/>

Instalación de Docker compose:

<https://www.digitalocean.com/community/tutorials/how-to-install-docker-compose-on-ubuntu-16-04>

Quagga y otros software OpenSource para ruteo:

<https://keepingitclassless.net/2015/05/open-source-routing-comparison/>

Strapi: <https://hub.docker.com/r/strapi/strapi/>

Strapi + Nginx: <https://blog.strapi.io/how-to-deploy-a-strapi-application/>

Consignas

Configuración de Red

- 1.- Aplicando VLSM, calcular las redes IPv4 para cada sistema autónomo según la información provista por los profesores.
- 2.- Trabajar con un nombre de dominio único para cada sistema autónomo, según la información provista por los profesores.
- 3.- Completar la tabla de asignación de redes que se lista más arriba.
- 4.- Configurar los sistemas autónomos con BGP.
- 5.- Interconectar los sistemas autónomos con al menos otro grupo de trabajo.
- 6.- Configurar OSPF como protocolo de ruteo interno dentro de cada sistema autónomo.

Servidores

- 7.- Instalar en su workstation el siguiente software:
 - VirtualBox
 - kubectl. Configurar alias para el comando kubectl a "k"
 - helm
 - ansible
- 8.- Importar y/o crear servidores en VirtualBox. Asegurarse de tener configurado adaptador puente.
- 8.1.- Importar microk8s: <https://bit.ly/2PIELCd>
- 9.- Configurar y/o usar usuario "ubuntu" con contraseña "ubuntu".
- 10.- Configurar IP estática en el servidor (netplan).
- 11.- Configurar acceso SSH utilizando clave pública/privada. Todas las acciones sobre los servidores deben ser llevadas a cabo con el comando SSH. Puede implementarlo con Ansible (opcional).

DNS

- 12.- Instalar software de DNS (Bind9, pero puede incluir Webmin)
- 13.- DNS Autoritativo: Configurar DNS en cada sistema autónomo y asignar registro A y PTR a cada dirección IP asignada.
- 14.- DNS Recursivo: Configurar la solución de manera tal que desde cualquier host sea posible resolver cualquier nombre de dominio.

Kubernetes

- 15.- Asegurarse que Kubernetes está instalado y los servicios funcionando:
 - microk8s.start
 - microk8s.enable storage dns ingress
- 16.- Importar la configuración de K8s en su workstation creando el archivo ~/.kube/config con el siguiente contenido. Reemplazar la dirección IPv4 con la asignada a su servidor K8s.

```
apiVersion: v1
clusters:
- cluster:
```

```

certificate-authority-data:
LS0tLS1CRUdJTiBDRVJUSUZJQ0FURS0tLS0tCk1JSUMvakNDQWVhZ0F3SUJBZ0lKQ
VBnMXZ6eUQ2RVl5TUeWRONTcUdTSWIZRFFFQkN3VUFNQ1F4RWpBUUJnTlYkQkFNTU
NURXlOeTR3TGpBdU1UQWVGdzB4TlRBME1qZ3lNekkyTVRSYUZ3MDBOake1TVRNeU1
6STJNVFJhTUJReApFakFRQmdOVk1JBTU1DVEV5Tnkd0xqQXVNVENDQVNjd0RRWUpL
b1pJaHZjTkFRRUJCUUFEZ2dFUEFEQ0NBUEw9DCmdnRUJBTUM2MUVOMUFRtGNazlo0Q
UpPb3p4b051ME9tZWZhU1l1K3FSM0pKS3ZJWmdlSlN2TnNJeWUySjJlNWcKTGFUUn
p4NUJHZ29WZ2wzcTZlOU83OW1sL0hPYlNUeGJiWmJGV1lWT1J3emVHa1R2OGVxSSs
2NlhJR0srSVdxbgpcWp6Tj1URm9DMkh3MVZLTHA1SFU5Qmc1SW9uRUdnWTNzZjNQ
L1RFQlVpL2EzWXJIN2dBS3pvN2pYK0tWRDcxCKNGZVlKZnVrU1YxWHVlVi90bmhKa
GJpcmN4TEpJNy9qK3djdHRvc1RkNEFlUmdJeG9uNjRSWlgzZXc0RGFDZ0QKekdzMk
w0T1JhalNEVlV0V0dsemZIRjAzNWhGclGNHNuVkpSSUdSNTVabERpVGZEU24wRk1
2NGVYd094Q0tHNQpnRkN1dnptTGxjN1l5UTBEZWJqZ0pka01tVkdVQXdfQUFhTlRN
Rkv3SFFZRFZSME9CQ1lFRkNHUWRJb0NGN2VNCno5VGNTWHBZaWpNUGMzaVNNQjhHQ
TFVZE13UVlNQmFBRkNHUWRJb0NGN2VNejlUY1NYcFlpak1QYzNpU01BOEcKQTFVZE
V3RUIvd1FGTUfNQkFmOHdEUVlKS29aSW9uRUdnWTNzZjNQZ0pka01tVkdVQXdfQUFh
TaXlhWTRnWk5OegpDZXUxLys0YU1CU0xwQ3V1UGtpMXZYOVBoWmdNSnpVaWVCZHJJ
NmhuUkYzZWJVK1JVaXR5NURXSnr3cDVCODhSCjZnOElsK1d0dE1OZC8xY3hqGxxV
nJtZXNCVm1YRVhCcFAwZEpkYnk0aXFmajhEazJoa0ljMU5uaDBjaTRKTGcKOTThwaU
5vQ1VCYTZVdUxUZ0daT2lWazdTmk1lOG9neXFSUjhaTEFlOVpTQUt3azJXU1E0OHF
EU1JLN1hwcEhTMgpXQWRISDF1OVZUEFuc1A3QkxGQ1o5ejRFSng0UXpTb2tGYSta
cG9CWGFBEkdUQXdxUDFwSkU5NGFpelVnbUfMck4xQ3Uva2V4YWZlcE9NV0JQS2Q2c
Gt2aVFEdEp6WUZpbTNMQmNINlM3Snc1dzJ3THZYSNsMUt2Q09oQ1B6RFUKZWRRVQ
otLS0tLUVORCBDRVJUSUZJQ0FURS0tLS0tCg==
server: https://192.168.0.10:16443
name: microk8s-cluster
contexts:
- context:
    cluster: microk8s-cluster
    user: admin
name: microk8s
current-context: microk8s
kind: Config
preferences: {}
users:
- name: admin
  user:
    password:
eXVvZmlQWnI2SWWhPMjV5cU1ERXRFZ1lVWV51Wk5yUkIycTdsMzk2cTVJcz0K
username: admin

```

17.- Instalar helm tiller en Kubernetes utilizando el comando 'helm init'.

Aplicación Web

18.- Desarrollar una aplicación web utilizando Strapi (<https://strapi.io>). La aplicación debe ser de inventario de recursos de red. Se debe poder crear un registro de equipo de red con su valor de número de serie y un booleano de si está o no conectado a la red. Se debe poder actualizar su valor y se debe poder eliminar un equipo ya existente. Todas estas acciones se deben ejecutar de manera autenticada.

19.- Usar github como repositorio para el desarrollo de la aplicación. A través de docker hub configurar un *automated build* para que cada vez que se suba un commit al repositorio se construya automáticamente la imagen de Strapi y se publique en docker hub. Declarar cuál es el hash de la imagen de docker generada.

20.- Desplegar la aplicación web sobre Kubernetes:

- Strapi (deployment)
- Mongo (stateful set)
- Services
- Ingress