LAB 9

CELL (Réseaux cellulaires)
Prof. Serge FDIDA
Prof. Mai Trang NGUYEN

Lab Assistants: Dimitris Kefalas, Sokratis Christakis
(Dimitrios.Kefalas@lip6.fr, Sokratis.Christakis@lip6.fr)

Subject: Deploy 5G real-world networks using K8s and SLICES Blueprint
Date: 02/12/2024

In this lab, we explore **SLICES**, a research platform designed to support experimentation and innovation in 5G and beyond-5G networks. The SLICES framework uses **blueprints** to define how 5G systems should be deployed and tested, ensuring consistency, reproducibility, and adaptability for different research needs.

- ☐ v2 of the SLICES Blueprint (revised version) can be found in the following link: https://gitlab.noc.onelab.eu/onelab/slices-5g-blueprint/-/blob/main/5G_deployment_revised.md
- ☐ v3 of the SLICES Blueprint can be found in the following link: https://doc.slices-ri.eu/BlueprintServices/beyond5G/beyond5G.html

In particular, Blueprint v3 (BP3) extends (among several features) Blueprint v2 (BP2) by adding the RIC (RAN Intelligent Controller) functionality, enabling enhanced control and optimization of the RAN through real-time analytics and decision-making.

Why SLICES Matters

By experimenting with these blueprints, we gain insight into real-world 5G challenges, such as:

- Optimizing resource allocation in single or multi-node clusters.
- Understanding how to separate or integrate the 5G Core and RAN based on deployment goals.
- Testing innovative technologies like radio emulators or software-defined radios for future network advancements.

5G Blueprint Deployment Scenarios

The blueprint outlines two primary deployment scenarios:

1. **Single Node-Cluster Deployment**:
    - Here, the **5G Core** and the **Radio Access Network (RAN)** are co-located on the same physical node.
    - This setup simplifies deployment and is often used for small-scale experiments.
    -
2. **Multi-Node Single-Cluster Deployment**:
    - In this scenario, the 5G Core and RAN are deployed on separate nodes, which could be in the same facility or across different locations.
    - Interconnection is established using options like dedicated links, research network backbones, or even the public Internet with VPNs.
    - This configuration mimics real-world, large-scale deployments where the core and RAN operate independently but must communicate efficiently.

For the purposes of this lab we will focus only on the first scenario (**Single Node-Cluster Deployment**) where the 5G Core and RAN are co-located on the same physical node.

FlexRIC with SLICES Blueprint

The **SLICES blueprint** automates the deployment of **FlexRIC** using **Ansible** and **Kubernetes**, streamlining the setup of 5G network components for research.

1. **Ansible Playbooks**: Automates the installation and configuration of **FlexRIC**, **Near-RT RIC**, and **RAN**, ensuring consistent and error-free deployment.
2. **Kubernetes Cluster**: First, the blueprint deploys a **Kubernetes cluster**, managing resources and ensuring scalability for all components.
3. **5G Core Deployment**: The **5G Core** (AMF, SMF, UPF, etc.) is then automatically deployed to manage user equipment and session control.
4. **gNB and UE Deployment**: The **gNB** (radio base station) and **UE** (user equipment) are deployed to simulate real-world 5G connectivity.
5. **FlexRIC Deployment**: **FlexRIC** is deployed to optimize and control the **RAN** in real-time using the **E2 interface**, ensuring dynamic network optimization without manual intervention.

The blueprint supports both **single-node** and **multi-node** deployments, making it flexible for different experimental setups. This automated process makes deploying FlexRIC efficient, scalable, and reproducible, ensuring consistent results for 5G research.

Lab Setup: Deploying 5G Core, RAN, FlexRIC, and Monitoring xApp

In this lab, you will deploy the 5G Core, Radio Access Network (RAN), FlexRIC, and a Monitoring xApp using Kubernetes, Ansible, and FlexRIC. The process is automated through pre-configured playbooks, ensuring an efficient and reproducible environment. Follow the steps below to set up the entire environment and monitor the network.

Step 1: Clone the Repository

Clone the repository containing the necessary files and configurations to your Virtual Machines (VMs):

```
git clone
https://gitlab.noc.onelab.eu/dkefalas/labs_cell.git
```

Step 2: Verify Your Kubernetes Environment

Before proceeding, ensure that you have a functional Kubernetes environment:

- If Kubernetes is already functional, proceed to the next step (Ansible installation).
- If Kubernetes is not set up or requires fresh installation, inform the teaching assistants for help.

Step 3: Proceed to Ansible Installation

Once your Kubernetes environment is ready, proceed with the Ansible installation:

1. Update the `hosts.yml` file:
   Modify the `inventories/UTH/hosts.yml` file with the correct IP addresses and node names of your VMs.
2. Modify the `group_vars/all` file:
   Set the appropriate ansible_user value.
3. Add the VM's own SSH public key to its `authorized_keys` file to allow SSH access to itself via `localhost`.

   ```
   cat ~/.ssh/id_rsa.pub >> ~/.ssh/authorized_keys
   ```
4. Install the necessary kubernetes extension for the ansible.

   ```
   ansible-galaxy collection install kubernetes.core
   ```
5. Deploy the scenario:
   Execute the following Ansible playbook to set up the 5G network components (Core, RAN, FlexRIC):

```
ansible-playbook -i inventories/UTH 5g.yaml --extra-vars "@params.oai-flexric.yaml"
```

Step 5: Verify Pod Deployment

Once the playbook completes, check that all required pods are running:

```
kubectl get pods -n blueprint
```

You should see output similar to:

```
NAME                               READY   STATUS    RESTARTS   AGE
oai-amf-6486c9d49c-snnt5           1/1     Running   0          45m
oai-ausf-84ffb6bc7c-vjzdj          1/1     Running   0          45m
oai-core-mysql-7f7b695b8b-s4l66    1/1     Running   0          45m
```

```
oai-flexric-5db68d7bf6-n28q8      1/1     Running   0          44m
oai-gnb-c5b4659c6-vghc7           1/1     Running   0          44m
oai-nr-ue-7dbdb954fc-4rbn4        1/1     Running   0          44m
oai-nrf-7dbb6d4b9-s9l2v           1/1     Running   0          45m
oai-smf-5d654698cf-7wpnh          1/1     Running   0          45m
oai-udm-7c49dc8f66-wsrtf          1/1     Running   0          45m
oai-udr-5d85996695-zg5g4          1/1     Running   0          45m
oai-upf-86dc5998c8-spzxz          1/1     Running   0          45m
```

Step 6: Verify UE to 5GCN Connectivity

Test the connectivity by pinging the 5G Core Network (5GCN) from the User Equipment (UE). Replace the pod name with the actual name from your setup:

```
kubectl exec -ti oai-nr-ue-7dbdb954fc-4rbn4 -n blueprint -- ping 12.1.1.1
```

Step 7: Deploy a Monitoring xApp

Now that your 5G network is running, deploy a monitoring xApp to observe Key Performance Metrics (KPM) for the RAN.

1. Connect to the RIC environment:

```
kubectl exec -ti oai-flexric-5db68d7bf6-n28q8 -n blueprint
```

2. Choose your programming language:
   Navigate to the appropriate directory for your chosen language:
   - For C:
     `cd /flexric/build/examples/xApp/c`
   - For Python3:
     `cd /flexric/build/examples/xApp/python3`
3. Run the xApp:
   To monitor KPM metrics, run the following command in the chosen language directory:

```
./xapp_kpm_moni
```

This xApp will collect real-time metrics for the RAN, including throughput, delay, and error rates.

Configuration Files and Directories

You can view and edit the configuration files for different components in the following directories:

- **Core Files:** `roles/flexric/files/blueprint/oai-flexric/core_files`
- **Core Configurations:** `roles/flexric/files/blueprint/oai-flexric/core_values`
- **RAN Files:** `roles/flexric/files/blueprint/oai-flexric/ran_files`
- **RAN Configurations:** `roles/flexric/files/blueprint/oai-flexric/ran_values`
- **FlexRIC Files:** `roles/flexric/files/blueprint/oai-flexric/flexric_files`
- **FlexRIC Configurations:** `roles/flexric/files/blueprint/oai-flexric/flexric_values`

- **UE Files:** `roles/flexric/files/blueprint/oai-flexric/ue_files`
- **UE Configurations:** `roles/flexric/files/blueprint/oai-flexric/ue_values`

These files allow you to customize and configure the 5G Core, RAN, FlexRIC, and UE components based on your experiment's requirements.

By following these steps, you will have successfully deployed the 5G Core, RAN, FlexRIC, and a Monitoring xApp using Kubernetes and Ansible. This setup provides a fully functional 5G network for experimentation, allowing you to monitor and optimize network performance in real-time.

Task 1: Create an iPerf Client-Server to Test Your 5G Network

In this task, you'll set up an **iPerf server** in the **UE pod** and an **iPerf client** in the **UPF pod** to generate traffic and test network performance between the two.

Step 1: Set Up the iPerf Server in the UE Pod

Install and start the **iPerf server** in the **UE pod**, which will listen for incoming connections from the iPerf client.

Step 2: Set Up the iPerf Client in the UPF Pod

Install the **iPerf client** in the **UPF pod** and configure it to generate traffic towards the **UE pod**, with a maximum bandwidth of 10 Mbps.

Task 2: Monitor and Cross-Validate Throughput with xApp

In this task, you'll validate the throughput observed by the **iPerf server** and compare it with the output from the **xApp**.

Step 1: Monitor the xApp Output

Check the **xApp** for real-time network metrics after generating traffic using the **iPerf client**.

Step 2: Cross-Validate Throughput

Compare the throughput reported by the **iPerf server** and the **xApp** to ensure they align.