

Altice Training

Pre-Requirements

Elaborated by Devoteam for Altice Portugal - Out. 2025

Context

The requirements outlined in this document are necessary for the advanced training to be delivered by Devoteam to Altice Portugal. This training is scheduled for November 2025 and January 2026, and the requirements are organized by the specific content modules.

General requirements

As a general requirement, the user should be able to install WSL in their local computers. It requires Administrator privileges.

This tutorial assumes that the user's laptop is running Windows 11. Devoteam will not perform operating system (OS) troubleshooting during the sessions.

Open Windows PowerShell as Administrator and run:

```
wsl --install
```

Reboot your laptop.

```
wsl --install Ubuntu
```

Other requirements

Git

Install Git on WSL (Ubuntu). Already installed if the Ubuntu version is 24.04 LTS.

```
sudo apt-get update
sudo apt-get install git

git --version
```

GitHub

Create a GitHub Account.

[Creating an account on GitHub](#)

Visual Studio Code

Install Visual Studio Code.

[Visual Studio Code](#)

Ansible

Install Ansible on WSL (Ubuntu).

```
sudo apt-get update
sudo apt-get install software-properties-common
sudo add-apt-repository --yes --update ppa:ansible/ansible
sudo apt-get install ansible

ansible --version
```

Jenkins

Nothing to add. Jenkins will run using Docker on the local laptops.

Docker

Install Docker on WSL (Ubuntu) or use Docker Desktop if you've license for it.

[Install Docker Engine on Ubuntu](#)

- Add Docker's official GPG key:

```
sudo apt-get update
sudo apt-get install ca-certificates curl
sudo install -m 0755 -d /etc/apt/keyrings
sudo curl -fsSL https://download.docker.com/linux/ubuntu/gpg -o
/etc/apt/keyrings/docker.asc
sudo chmod a+r /etc/apt/keyrings/docker.asc
```

- Add the repository to Apt sources:

```
echo \
  "deb [arch=$(dpkg --print-architecture)
signed-by=/etc/apt/keyrings/docker.asc]
https://download.docker.com/linux/ubuntu \
  $(. /etc/os-release && echo "${UBUNTU_CODENAME:-$VERSION_CODENAME}")
stable" | \
  sudo tee /etc/apt/sources.list.d/docker.list > /dev/null
sudo apt-get update
```

- Install the latest version:

```
sudo apt-get install docker-ce docker-ce-cli containerd.io
docker-buildx-plugin docker-compose-plugin
```

- Add the \$USER to the Docker group.

```
sudo usermod -aG docker $USER
```

- Logout and Login on WSL. Test Docker (no sudo is required)

```
docker ps

docker run hello-world
Unable to find image 'hello-world:latest' locally
latest: Pulling from library/hello-world
17eec7bbc9d7: Pull complete
Digest:
sha256:6dc565aa630927052111f823c303948cf83670a3903ffa3849f1488ab517f891
Status: Downloaded newer image for hello-world:latest

Hello from Docker!
This message shows that your installation appears to be working correctly.
```

Minikube and Kubectl

- Minikube

[minikube start](#)

```
curl -LO https://storage.googleapis.com/minikube/releases/latest/minikube_latest_amd64.deb
sudo dpkg -i minikube_latest_amd64.deb
```

- Kubectl

[Install and Set Up kubectl on Linux | Kubernetes](#)

```
sudo apt-get update
sudo apt-get install -y apt-transport-https ca-certificates curl gnupg
```

```
curl -fsSL https://pkgs.k8s.io/core:/stable:/v1.34/deb/Release.key | sudo
gpg --dearmor -o /etc/apt/keyrings/kubernetes-apt-keyring.gpg

sudo chmod 644 /etc/apt/keyrings/kubernetes-apt-keyring.gpg
```

```
echo 'deb [signed-by=/etc/apt/keyrings/kubernetes-apt-keyring.gpg]  
https://pkgs.k8s.io/core:/stable:/v1.34/deb/ /' | sudo tee  
/etc/apt/sources.list.d/kubernetes.list  
  
sudo chmod 644 /etc/apt/sources.list.d/kubernetes.list
```

```
sudo apt-get update  
sudo apt-get install -y kubectl
```