

## The theme park analogy

# Kubernetes explained in pictures: the theme park analogy

MON, 9 JUL 2018

This is an attempt to explain Kubernetes core concepts (pods, nodes, services, resources, ...) with an theme park analogy.



Kubernetes (k8s from now on) comes with its own set of abstractions, its own lingo, but with a little bit of imagination we can correlate those abstractions with more familiar concepts.

I was hoping for the sausage factory analogy to work, but unfortunately k8s is more than sausages, so we will have to settle for a theme park analogy.

The following is based on [Google Kubernetes Engine](#); some of the concepts may work slightly

different in other providers or if you run k8s yourself.

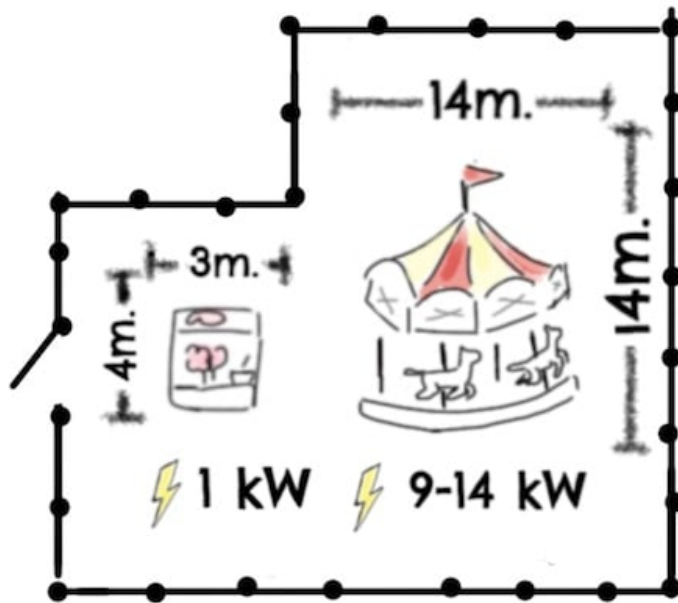
## Who are you

You are Kubenelius Fizzbuzz, a newly graduated MBA, with the speciality on Boxes Of All Sizes and you are about to embark on your greatest idea: KubePark Theme World.

## The fun ride plans

To start KubePark, you need to plan its fun rides. You decide that, for each attraction, you will just need to fill in a template like this (k8s [deployment](#)):

# THE 1909 CAROUSEL



x3

## Tags:

- carousel
- kid friendly
- classic

Here you say that KubePark will have three 1909 carousels.

A 1909 carousel is composed of two parts: a classic carousel plus a candy floss booth. Each part (k8s **container**) will be delivered by drone in a huge box, completely assembled and ready to roll once unboxed.

Each attraction will be surrounded by a fence, isolating it from other attractions, so that a problem in one attraction is not spread to other attractions. But the parts of the attraction form an inseparable whole (k8s **pod**), so there are no barriers or firewalls that could impede the communication or movement between the parts, ruining the enjoyment of visitors (k8s except if you want to **share files** which requires a little bit of configuration).

Having a fence means that you need to plan where the doors (k8s **containerPort**) for visitors should be. In the case of the 1909 carousel, visitors will come through the candy floss booth: a carousel ride is always more fun while enjoying a big fluffy sugar cloud.

The template also specifies the size (k8s memory) and power (k8s CPU) requirements (k8s **resources**), and tags (k8s **labels**) the 1909 carousel as a carousel, kid friendly and classic attraction. We will see later how you will use all this information.

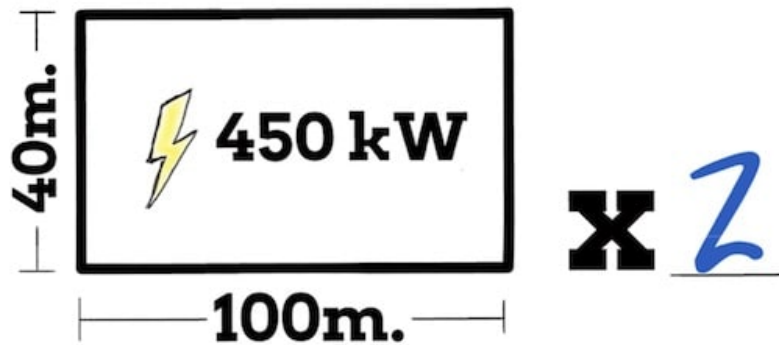
In the reverse of the template, not shown above, you write down the maintenance instructions (k8s **probes**) for each part. Your crew will periodically follow those instructions to know if any of the parts needs replacement. If it does, the maintenance crew will burn the faulty part to the ground and drones will bring another huge box with the replacement. So no fiddling around to try to fix anything, but just a plain simple replacement of the whole booth or carousel. During the replacement the attraction will be closed to visitors, even if some parts are still in a working state: the 1909 carousel experience must have candy floss and must include a carousel ride, both things are inseparable.

## The land

With the fun rides planned, it is time to build the park.

You sign a rent contract (k8s **node pool**) with a landlord:

**Contract:** KubePark basic plots



☒ **Free disaster insurance?**

**Tags:** Plains  
Mediterranean weather

**Signed by:**

A handwritten signature in blue ink, followed by a blue rectangular stamp containing the word 'plots'.



The contract is for two parcels of land (k8s **nodes**), each one with the specified size (k8s memory) and power generator (k8s CPU). Note that the contract is not for two particular parcels, but for any two parcels with those characteristics, it will be the landlord's job to find them. With the modern means of passenger transportation, they don't even need to be adjacent.

The landlord also takes care of securing the access to the parcel, by installing barbed wired around each parcel.

You also tick the free disaster insurance (k8s **auto repair**) box: if a parcel becomes unusable, the landlord will replace it for another one with exactly the same characteristics. It will be the landlord's sole responsibility to find out and decide if the parcel is unusable, and to replace it.

Lastly, any parcel provided by the contract will be tagged (k8s **labels again**) as plains with Mediterranean weather. We will see later how we use this information.

## Building KubePark

You already worked hard enough, so you delegate the details of the park's construction to your control crew. The crew will make sure that the exact number and type of attractions that you specified in your plans are always working.

In a gesture of unexpected generosity, your landlord offers an air traffic control tower (k8s **cluster**

master) to accommodate the control crew, and to take care of any maintenance and improvement work that the tower may need. One less thing to care about!

From the tower's privileged position and with your fun ride plans in hand, your crew will decide on which parcels to build each attraction, making sure that the parcel's available room and spare power can accommodate those required (k8s resource quality of service) by an attraction.

An attraction will always get the minimum power (k8s CPU) required, will never get more than the maximum power (as it will short circuit and burn otherwise), and will just get additional power (over the minimum) if there is spare in the parcel. If there is none spare, the attraction will just move more sluggishly.

The Conga Dancing Party Line has a minimum and maximum size (k8s memory). It will always get the minimum size specified, but if gets longer than the maximum size, it will be shutdown. Everybody knows that a too long a Conga always ends up in wild riots! It will get additional space (over the minimum) if there is spare in the parcel, but if there is none, an attraction that is over its minimum space will be shutdown. It is up to the traffic control tower to decide which one, which could be the Conga Dancing Party Line itself or another.

As there is plenty of room and power, the team has no problem creating all the attractions that you planned. The park is almost ready!

## KubePark opening

To help visitors reach the attraction you decide that it's easiest for them to use a coloured path wayfinder (k8s nodePort service) like:







Image attribution: [London Victoria station floor lines](#) by [Cmglee](#).

Here you have decided to create a blue lane that will take visitors to the attractions that you tagged as “classic” and a green lane that will take them to the attractions that you tagged as “roller coaster”. You can be as specific or generic as you need, and can combine multiple tags if necessary.

This wayfinder is not just for visitors, your internal crew could also use it, and you could even create paths visible to just your crew (k8s [clusterIP service](#)), for example to find the crew’s canteen.

As the parcels are completely surrounded by barbed wire, you need to somehow let the visitors to come into the park. One option would be to take one of the coloured paths all the way to the barbed wire and make a hole in it (k8s [loadBalancer service](#)), but this would mean that you will need to somehow secure each and every coloured path.

So instead you opt to have some admission gates (k8s [ingress](#)). Your landlord provides some standard gates, so that your crew do not need to maintain them, neither do they require space or power from your parcels. The landlord will also take care of building an overpass from the gates to the parcels. Maybe in the future you will opt for more specialized admission gates (k8s like [Traefik](#)).

You give the admission gates’ crew some simple instructions:

1. Visitors with kids younger than six, should use the red lane.
2. Visitors with just teenagers, should use the green lane.
3. ...

# Up and running!!!

And with this, your park is ready to roll:

