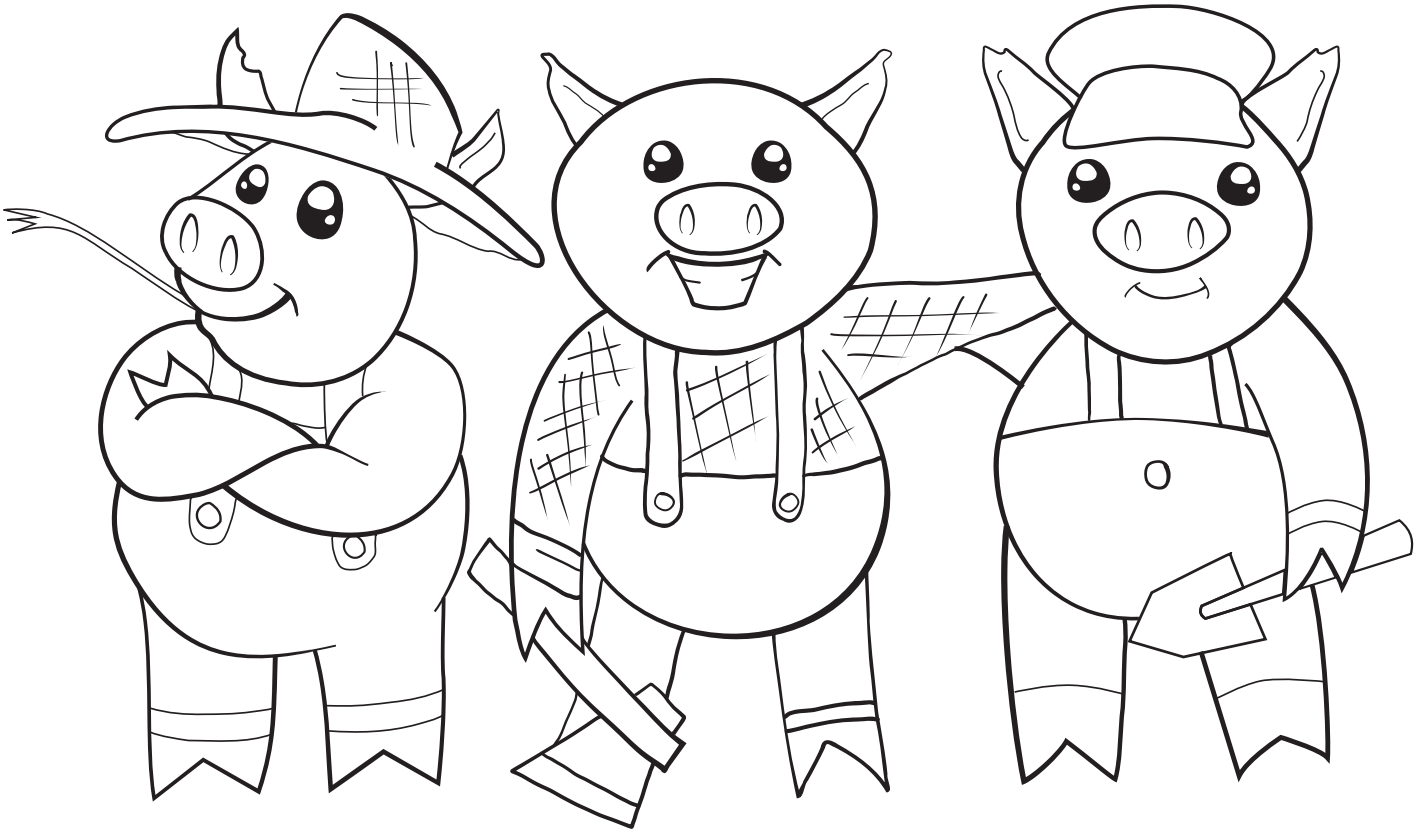
INTRODUCTION

Once upon a time, there were three little pigs. They each needed a place to live.

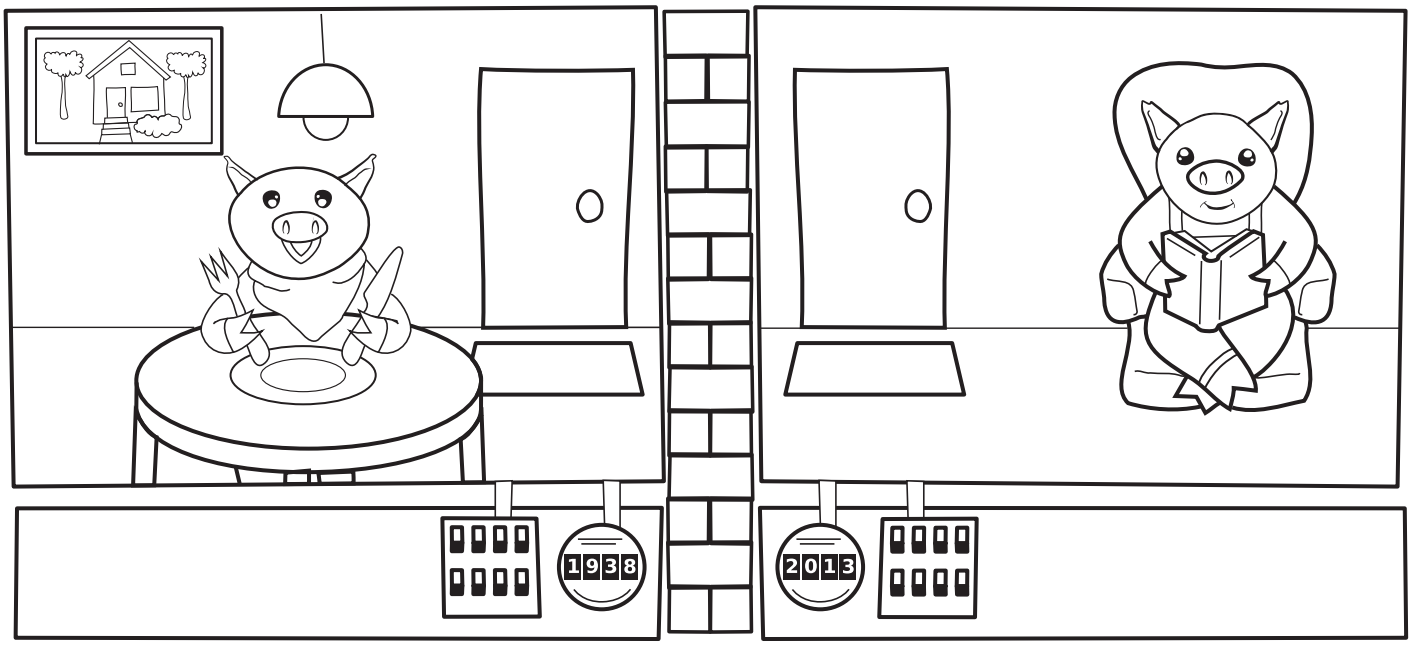


There's a lot of different types of places to choose from...

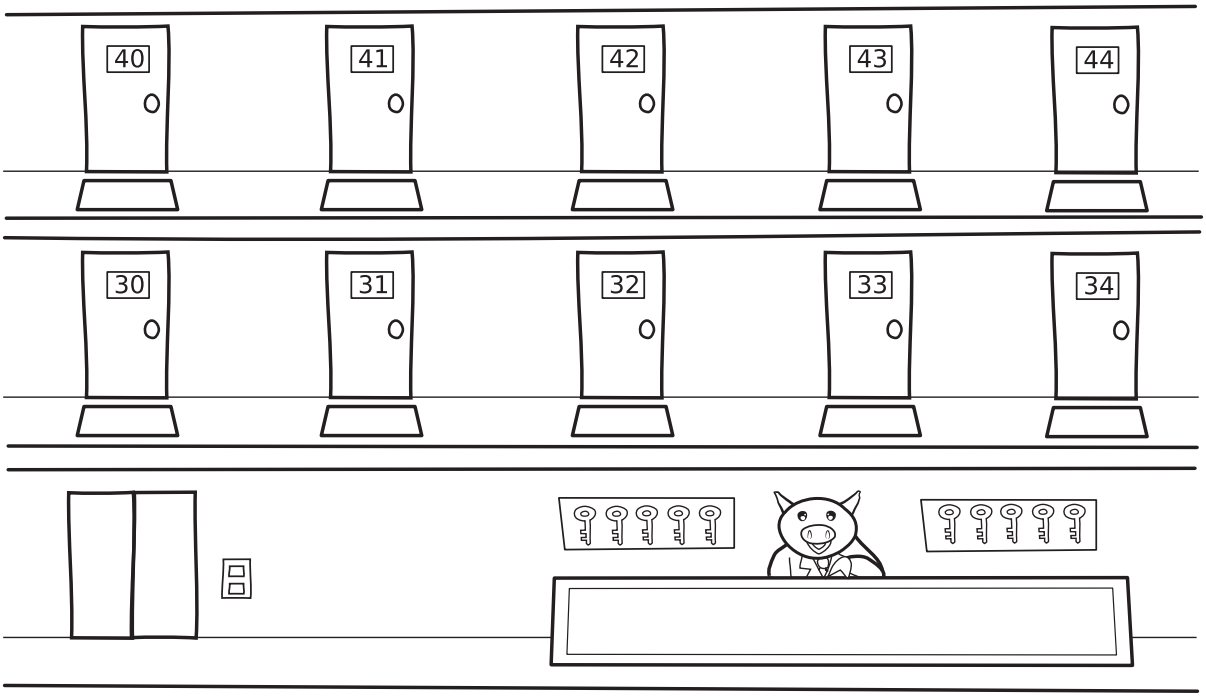
HOUSE DUPLEX APARTMENT HOSTEL PARK

If a piggy was an application.... living in a house (physical machine) would be the most secure. If one house is broken into, the other houses remain secure. A separate house per piggy means a lot more home maintenance, though!

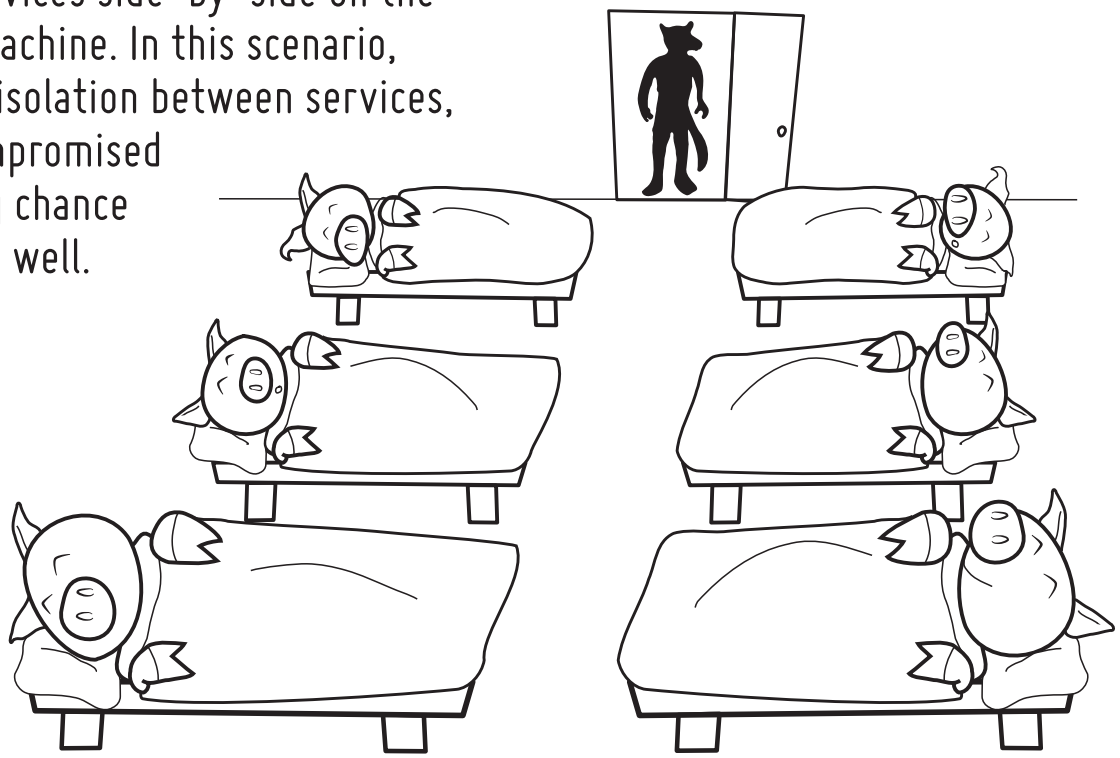
A piggy living in a duplex is like an application with multiple services deployed to multiple VMs on the same physical machine. While the structure is shared, the entry points are not. If one home is compromised, breaking in to the other VMs involves breaking through the hypervisor, sVirt, and the host kernel. However, you still have the costs of maintaining multiple OSes, with loss of speed and a limited ability to share resources.



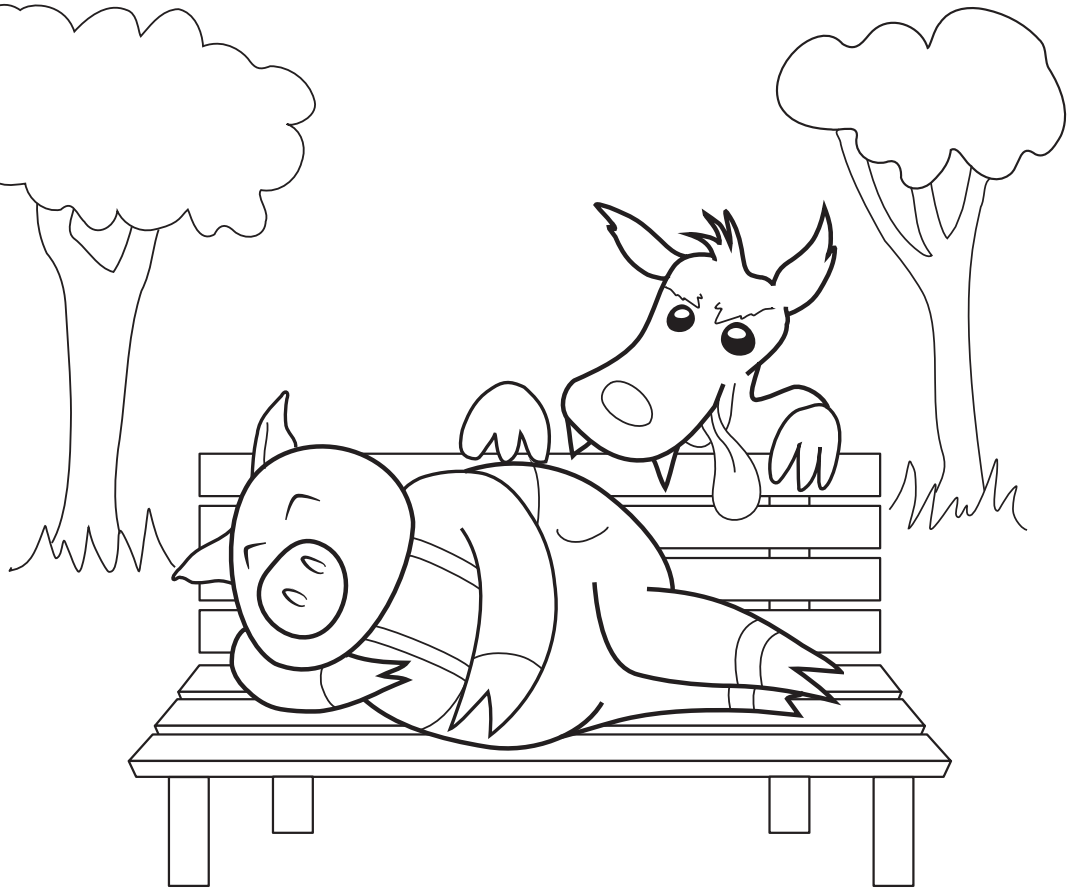
Piggies living in an apartment building are like applications running in containers. You get excellent sharing of services, lower cost of maintainence and decent separation. One problem, though, is that if the front desk were compromised, then all of the apartments would be compromised. This is similar to a container environment where, if the kernel were compromised, all of the containers would be as well.



Piggies living in a hostel are like running an application's services side-by-side on the same physical machine. In this scenario, there is limited isolation between services, but if one is compromised there is a strong chance others will be as well.
Of course, if you're running with SELinux, you'll have better isolation.

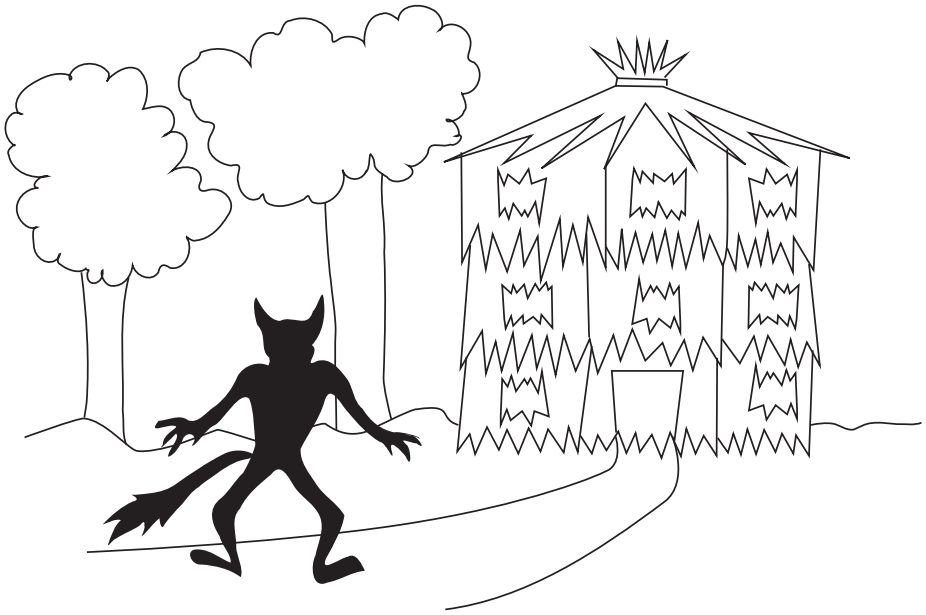


If they are up for living on the edge – as folks who run their apps on systems running setenforce 0 are – the piggy could consider sleeping in the park. We don't need to tell you how risky this is.



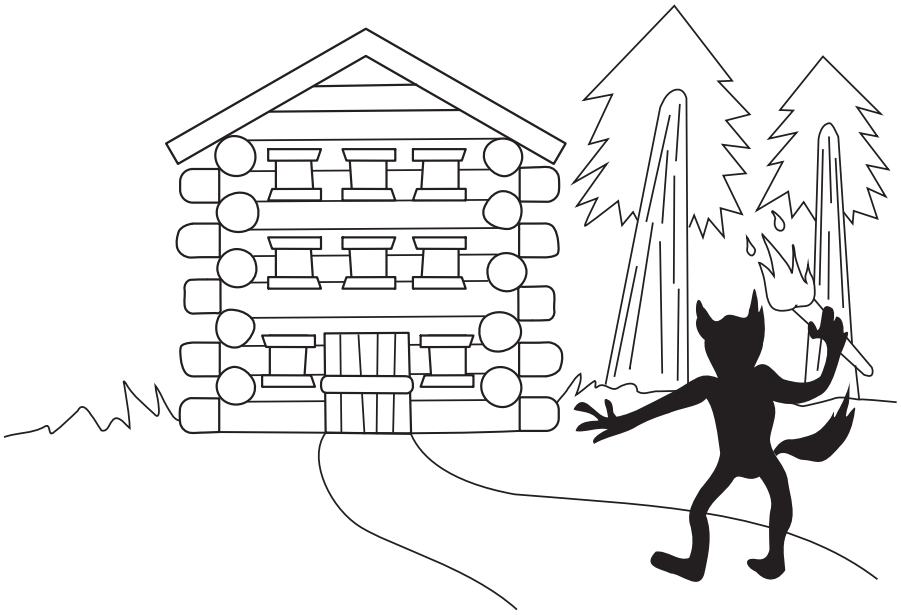
Containers, as represented by the apartment building, seem like a good middle ground. The apartment building offers better security than services sharing the same host, with more flexibility on content. Apartments provide better sharing of resources, startup speeds, and the cost of maintenance is lower than duplexes (VMs). Let's explore life at the apartment building in greater detail.

When choosing an apartment building to live in or a host platform to run your containers, construction quality is a top concern.



Running containers on a do-it-yourself platform is like choosing a piggy apartment building made of straw. Buildings made of straw require constant upkeep and you are on your own in terms of support.

Running containers on a community distro is like choosing a piggy apartment building made of sticks. It might be slightly more robust / reliable but still comes with no commercial support.



Running containers on a platform like Red Hat Enterprise Linux or OpenShift, Red Hat's container application platform, is like choosing a piggy apartment building made of brick. The platform is supported and maintained by a trusted partner.

Life in the brick apartment complex is best understood through the exploration of the following six characteristics...

1 NAMESPACES

2 RESOURCE CONTROL

3 SECURITY

4 IMAGES

5 OPEN STANDARDS

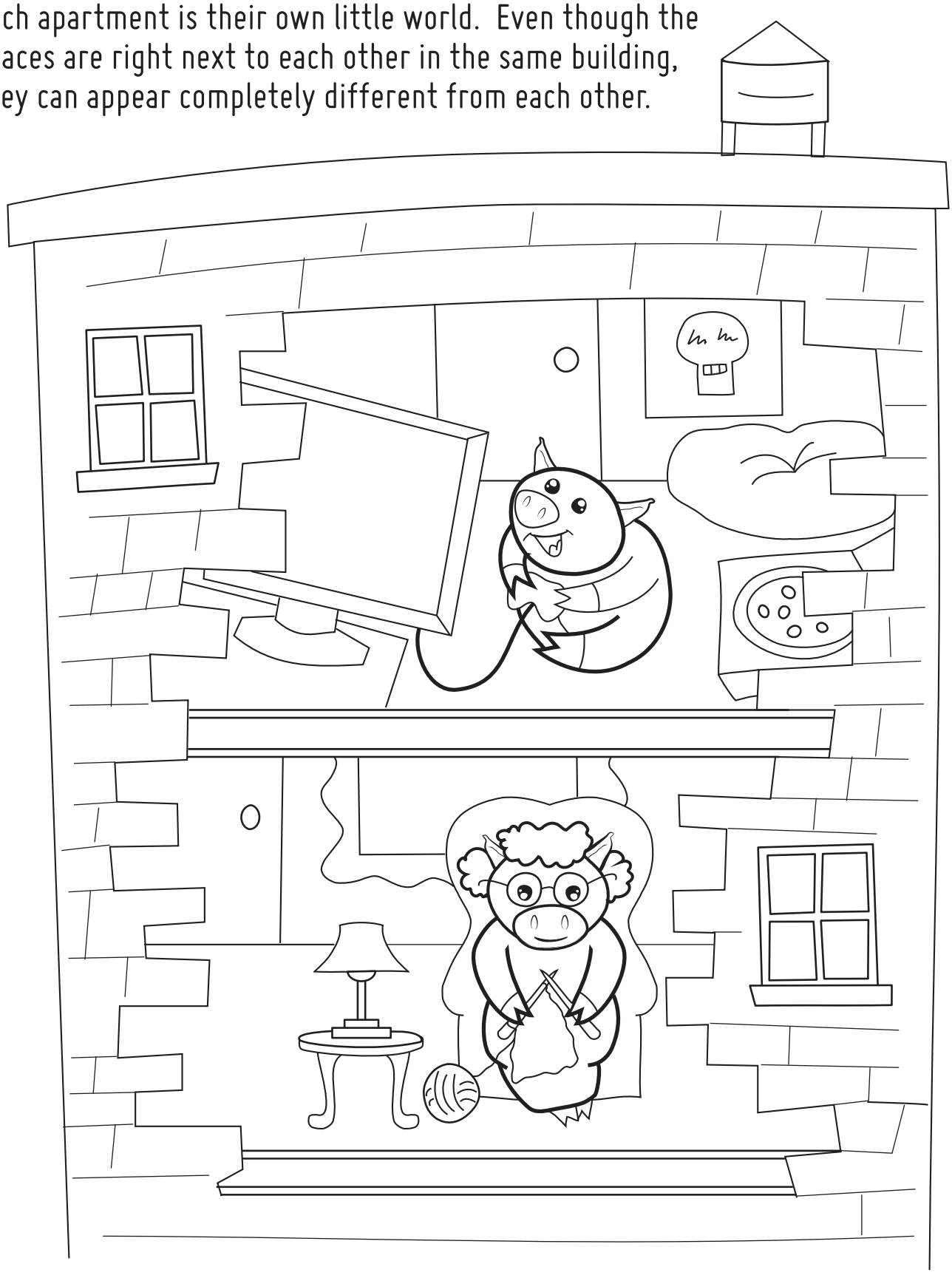
6 MANAGEMENT

NAMESPACES

Our piggy friends who live in apartments share the same building and basic layout. They personalize their space to make it their own.

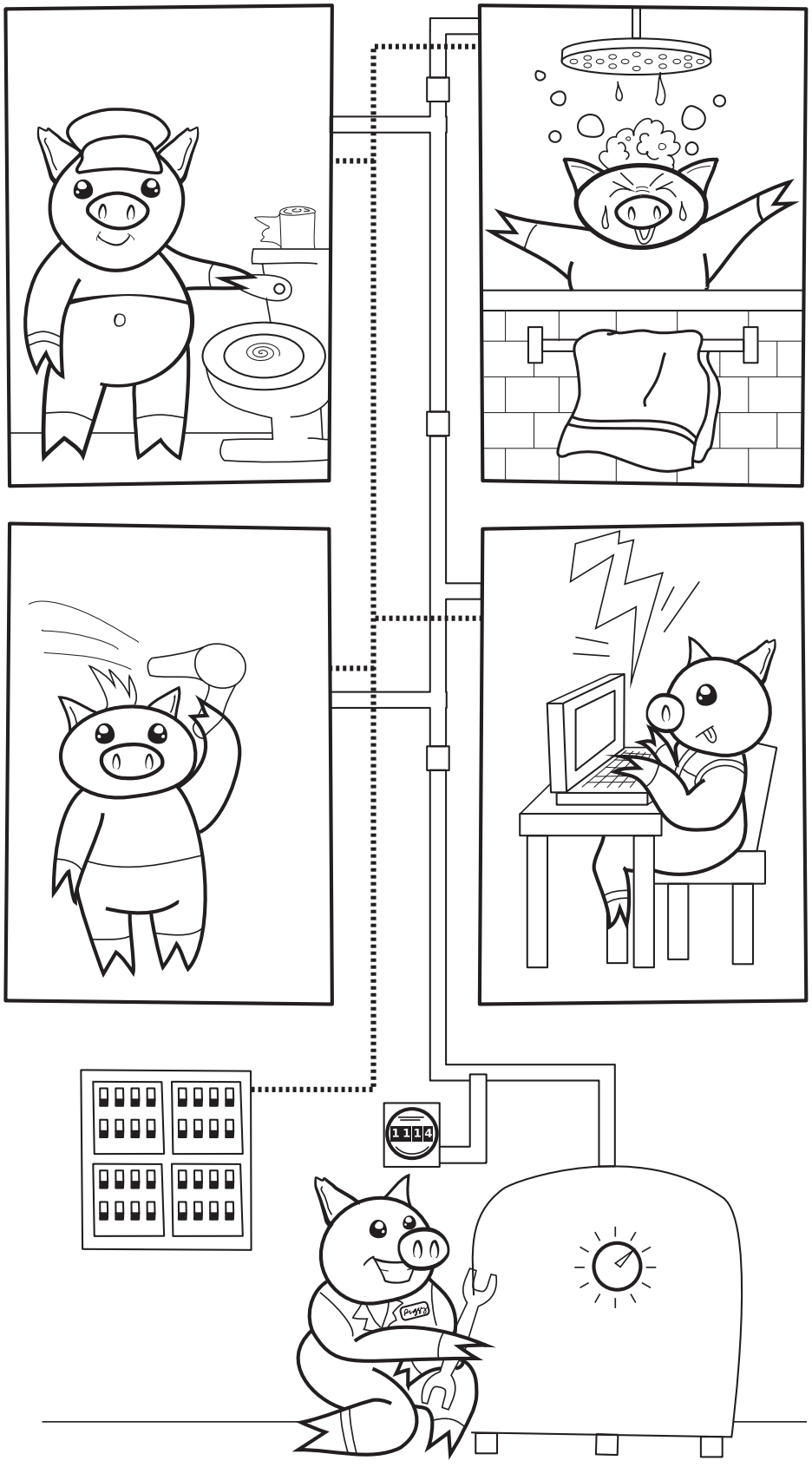
Container namespaces provide containers a way to identify and 'personalize' their own space (as the apartment piggies like to do.)

Each apartment is their own little world. Even though the spaces are right next to each other in the same building, they can appear completely different from each other.



RESOURCE CONTROL

In a shared resource situation, such as piggies sharing an apartment building, resource management is key to a good experience for everyone. For example, flushing the toilet in one apartment should not raise the water temperature in another. Blowing a fuse in one apartment should not kill the power in another.



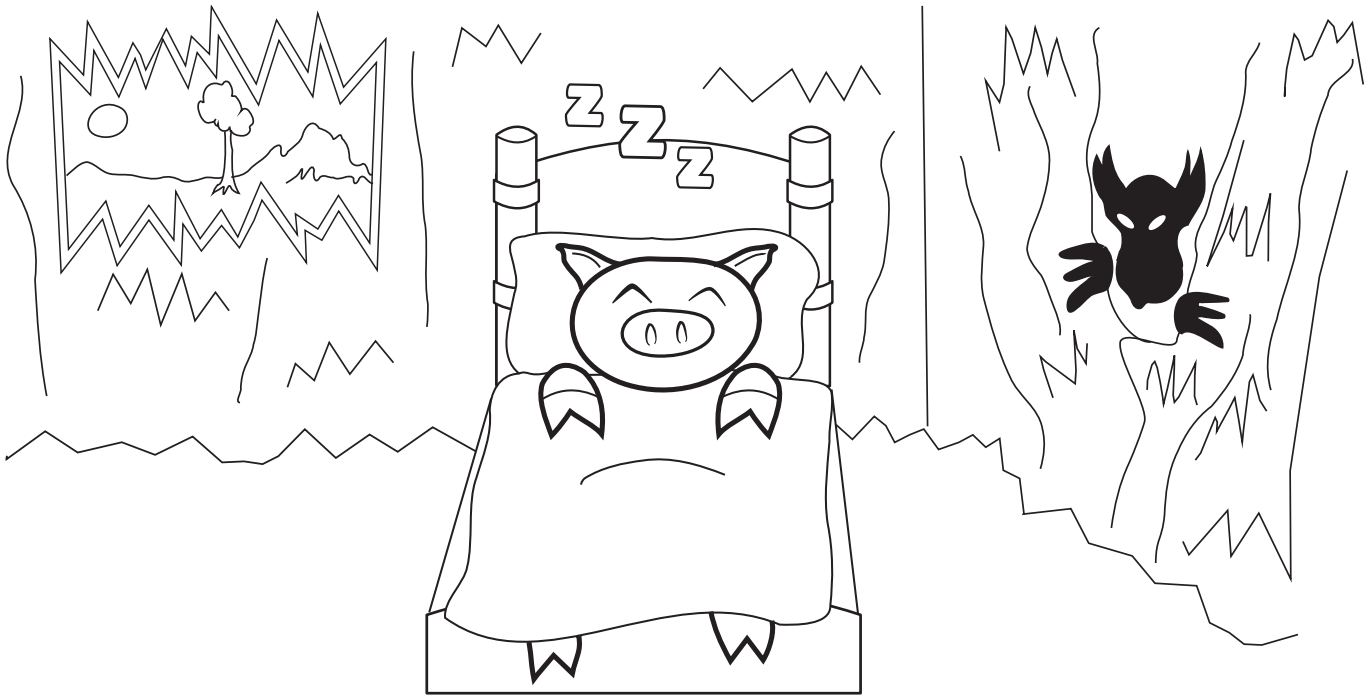
Cgroups are used to manage container resource control. If you have a poorly-written cgroup configuration, you'll run into problems with resources.

In the container world, you want the best performance for shared resources. You can rely on the Red Hat Enterprise Linux kernel for this.

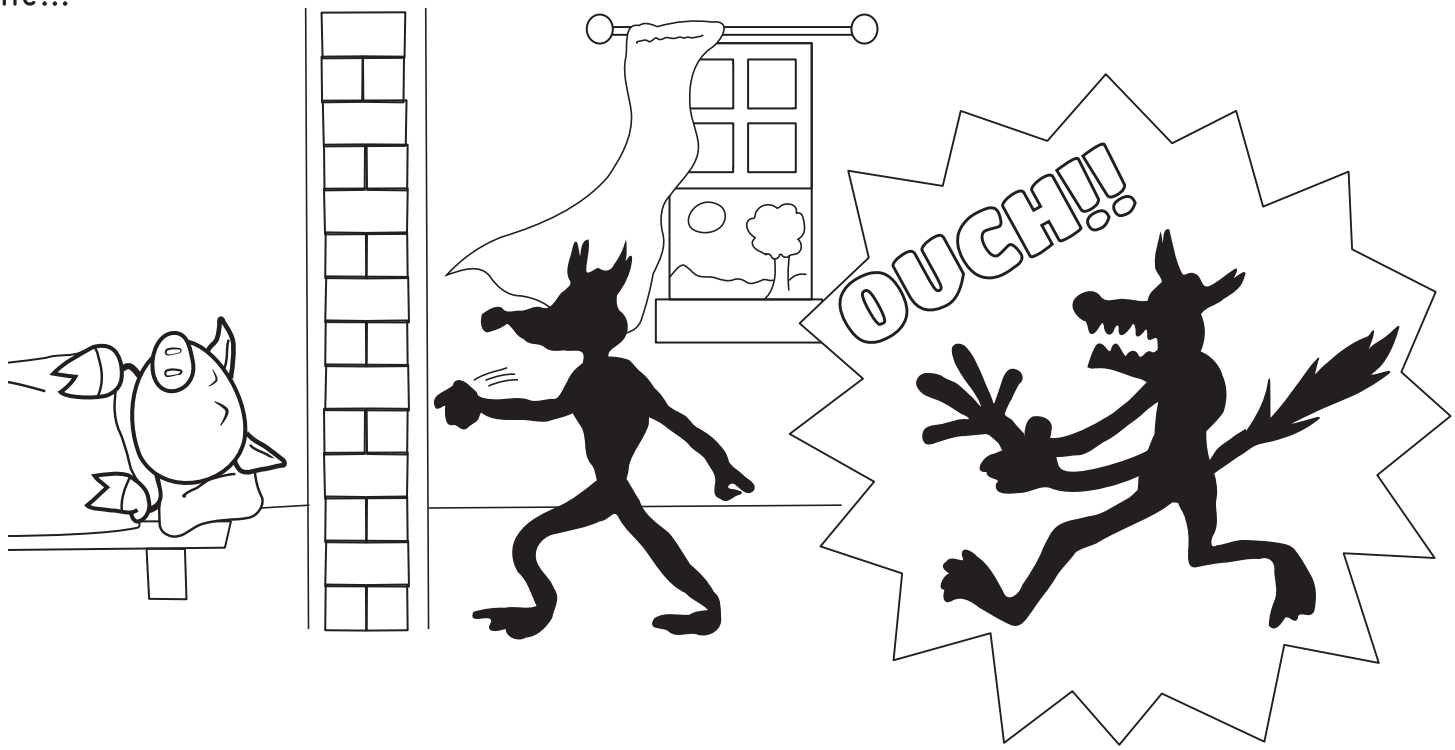
Think of a Red Hat subscription as access to the building super, who makes sure the infrastructure of the building is working correctly and who tunes it as needed.

SECURITY

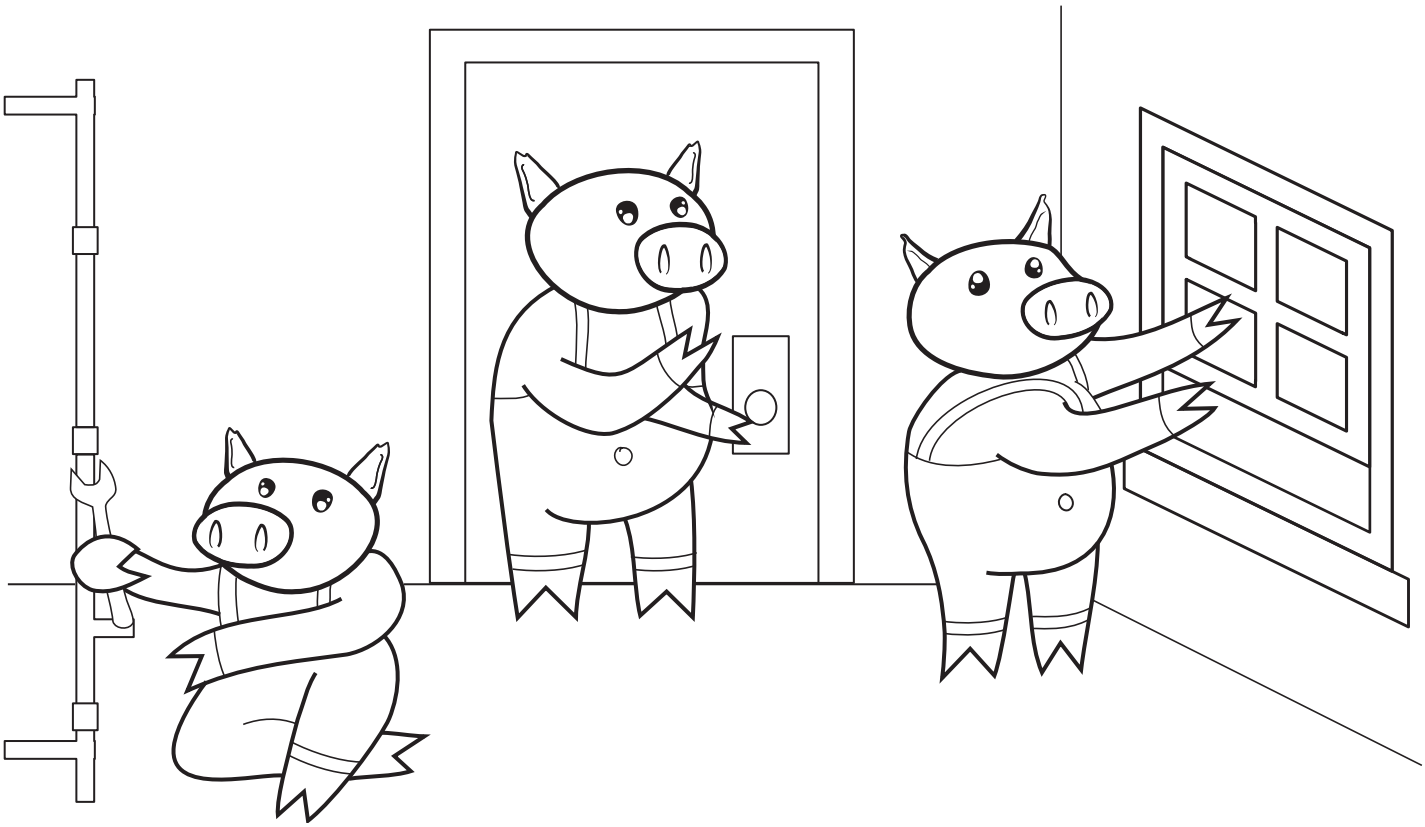
As with apartments, the most secure containers have strong walls between them. You don't want one compromised container to result in the whole system being compromised.



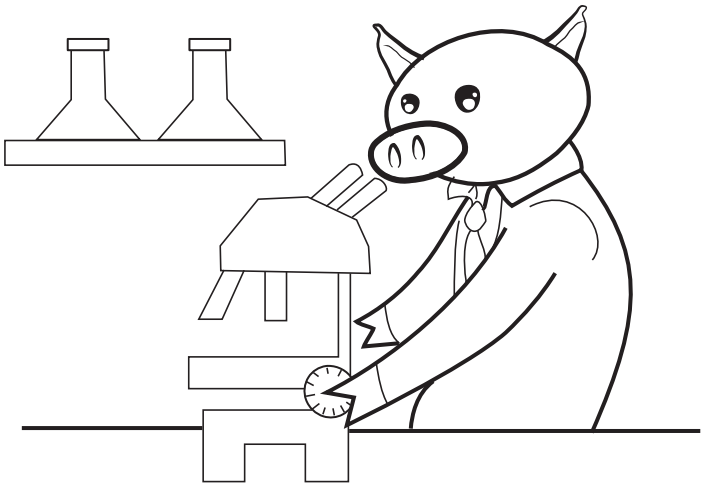
This is very important with containers, because the kernel is shared. What makes the Red Hat "Brick Apartment Building" more secure? SELinux, for one...



Your subscription also gives you access to security analysis tools (like Red Hat's Deep Container Inspection) to scan your containers and hosts for bad configurations and vulnerabilities...



... and access to a team of Red Hat security experts who fix issues as they arise.



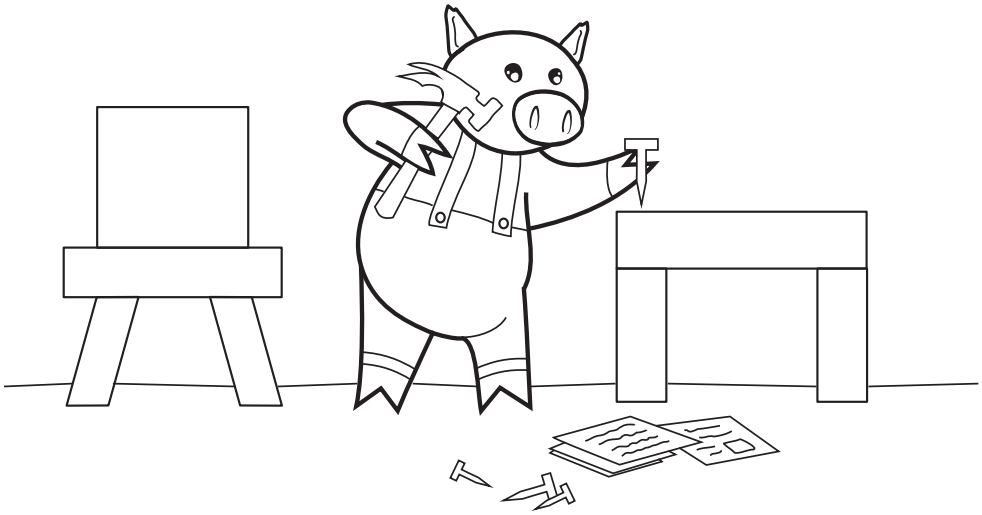
Good security practices lower a piggy's risk of an unexpected roast!

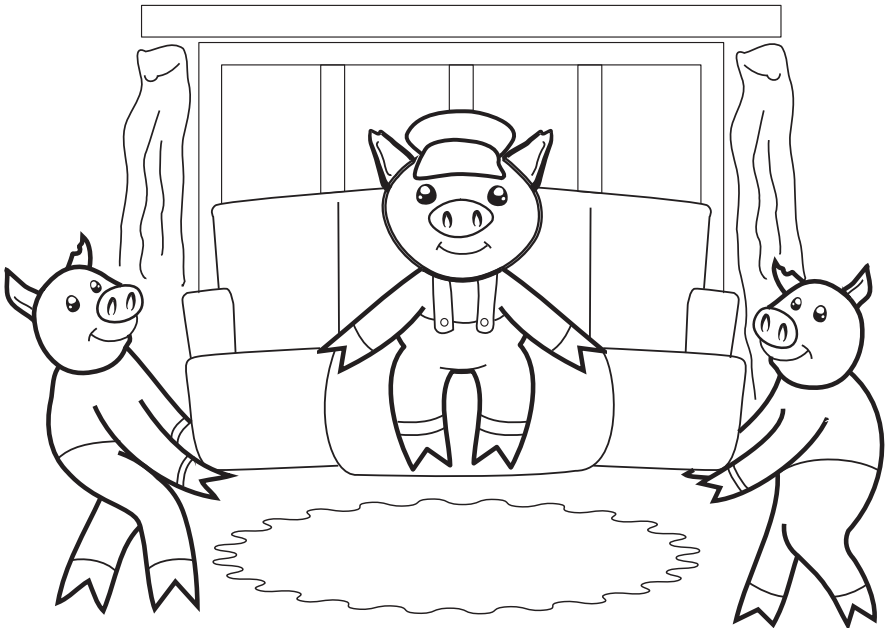
IMAGES

It can be overwhelming to furnish an empty apartment (or container) from scratch.

This piggy sourced some furniture curbside - the safety and cleanliness of such finds is somewhat questionable... almost like picking random container images off the Internet.

This piggy picked up furniture pieces at a warehouse to assemble himself. Pain-staking and time-consuming... almost like building your own base container images.



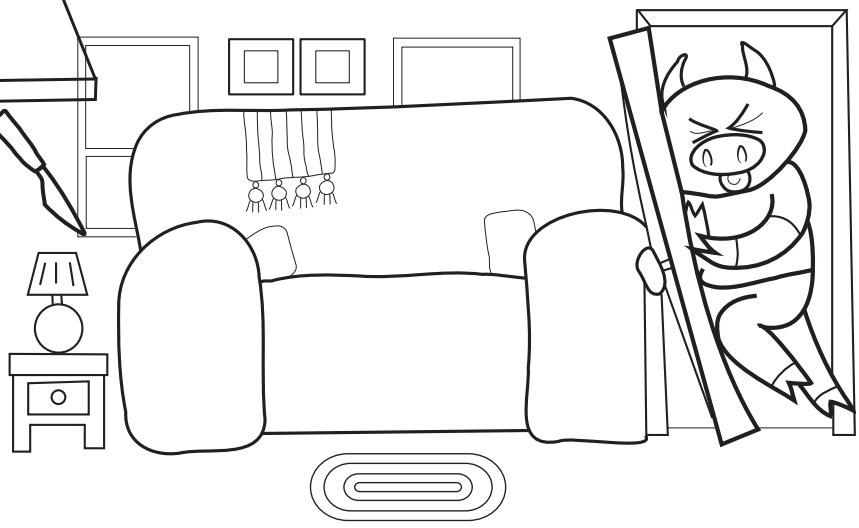


This piggy purchased high-quality, factory-assembled furniture from a showroom and it was delivered to his home via white-glove service. This is like downloading Red Hat certified container images from the Red Hat Registry or from your local Satellite Server.

COMMUNITY STANDARDS

When selecting a piggy apartment building, it's important to ensure that its infrastructure is compliant with common industry standards and policies.

What if your appliances run at a different voltage than what is provided in your new apartment? You may need to repurchase a number of expensive appliances (or rearchitect your applications).



If your furniture is too large (or too small), living in the apartment might require some amount of adjustment.

Standardization and consistency create a common foundation that leads to greater application portability. At Red Hat we always attempt to work with the upstream first. In containers we are the #1 contributor to Docker other than Docker, Inc and #2 in Kubernetes to Google. We also work with the Open Container Initiative and the Cloud Native Computing Foundation to help set and promote shared standards.

Whether it's piggy apartments or Linux containers – infrastructure consistency means you can confidently deploy container-based applications anywhere, from bare metal to cloud environments.

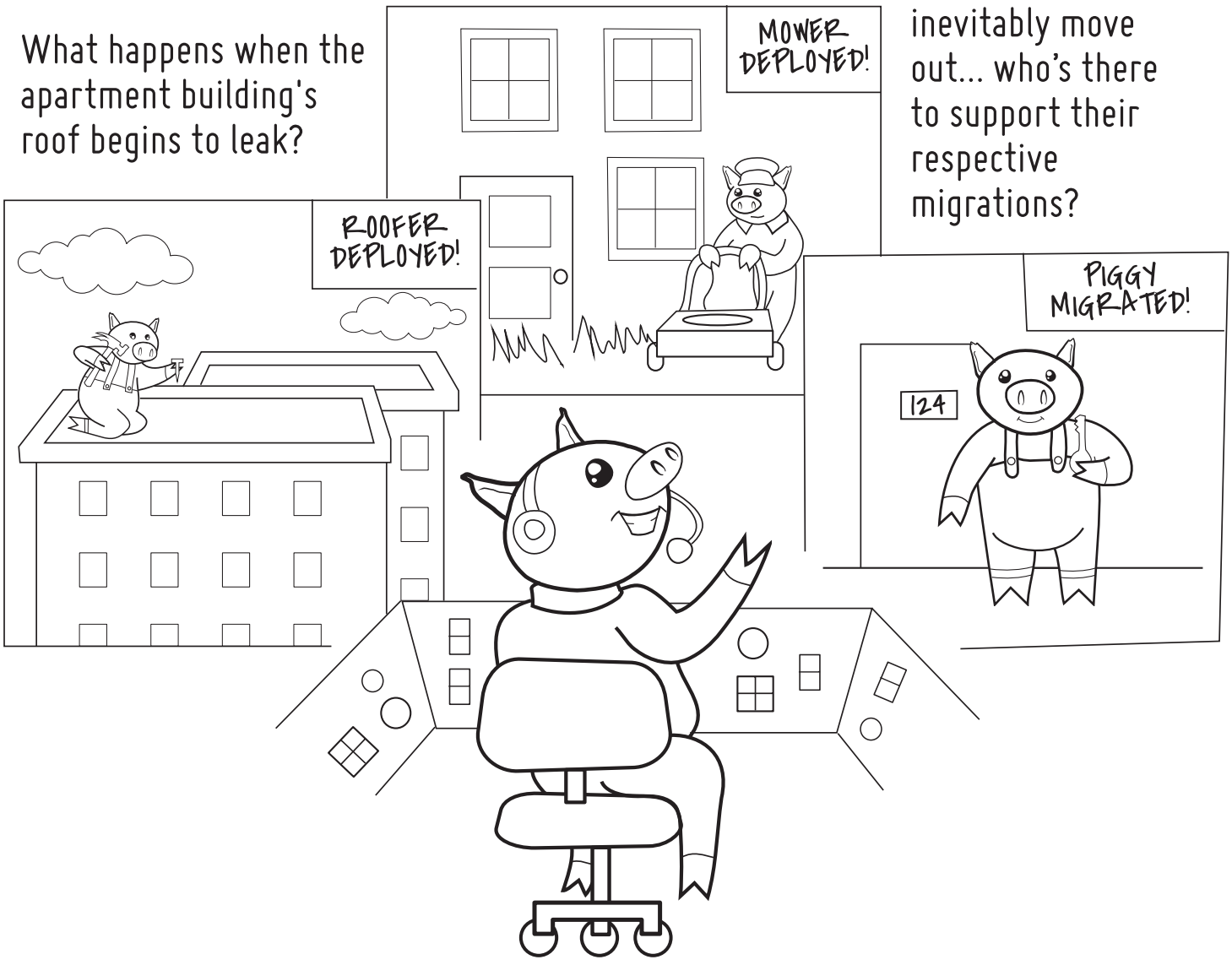
MANAGEMENT

As you expand to house many piggies across many apartment buildings, management and upkeep quickly become complicated and time consuming.

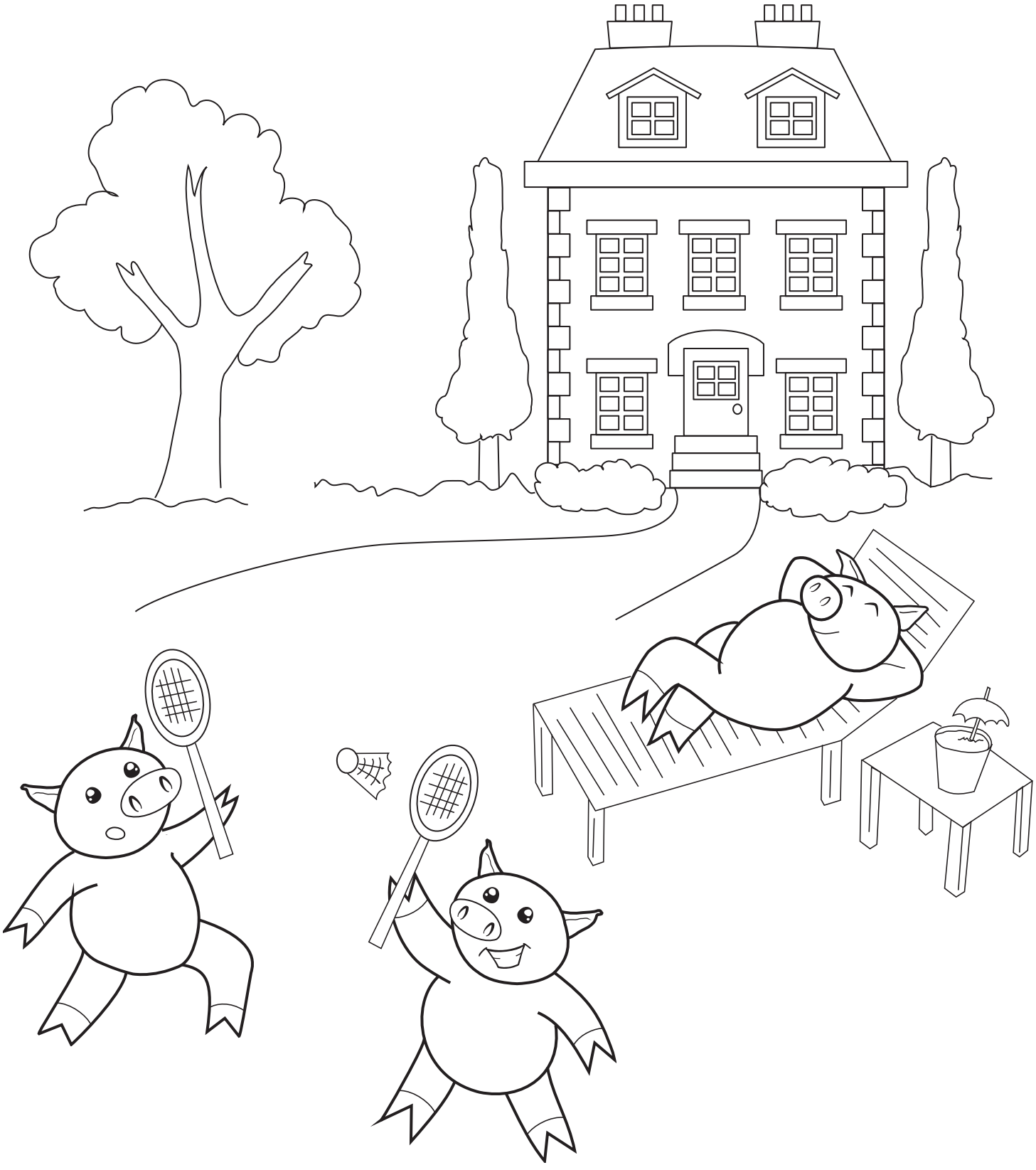
What happens when the apartment building's roof begins to leak?

What happens when the lawn becomes overgrown?

When new piggies move in and others inevitably move out... who's there to support their respective migrations?



Management and upkeep is important with apartments and apartment buildings – especially as you scale up. The same is true for application containers. OpenShift, Red Hat's container platform, works in concert with Red Hat CloudForms to help you streamline node and container creation, deployment, orchestration workflows, and management.

THE END

The piggies have finally found their perfect home. Ready to make the move? Visit **http://red.ht/containers** to learn more.