How containerization helps easy deployment of secure infrastructure and staff training to new onboarding clients.

Table of contents

How containerization helps easy deployment of secure infrastructure and staff training to new onboarding clients. 1

Introduction 3

How it works and why 4

The stack 5

How we configure and secure the containers 6

How we deploy and orchestrate 7

How we train 8

Conclusion 9

Introduction

It used to be a given that businesses had their own secure infrastructure needs. Those arose from their hardware, from routers, switches, firewalls and modems all the way back to internal monitoring systems. Also, one of the key factors were each business´ server hardware (or software and OS) and applications running there, as well as their security engineers on board, with their differing sets of capabilities, preferences and best practices combined.

Nowadays, we are effectively in an age of secure infrastructure, that can be easily built, configured, deployed and scaled with similar results across businesses.

This is achieved by containerization of the infrastructure, with simple turn-key configuration, where each business practically only checks their needs with the setup options while setting up the container.

It is a robust solution for both those with smaller demand of routes and subnets, number of devices or apps - and those with large multiscale networks, server clusters and both legacy and new production software on board. Therefore containerized infrastructure is viable option for medium businesses and enterprise alike.

How it works and why

It is only fair to mention upfront - migrating your existing security core practices to the whole new architecture can be seen as risky and expensive.

But when negotiating with your account and IT team, one benefit stands among all - by migrating to containerized secure infrastructure, you aquire not only industry standard infrastructure, but also one that works from the get-go, requires much less maintenance (made even easier by automating tasks) and is failsafe thanks to redundancy, load balancing, auto-updates, easy disaster recovery and many more features built into the core of it.

In the (not so) long run, your infrastructure not only gets more robust, but also requires much less manpower to maintain - which will further mitigate the initial arguable pain points.

We are firm believers that we provide real value to businesses this way - as we not only strive to provide the most secure platform, but also one that does not need our constant presence on the payroll to run smooth.

In fact, once set up correctly and with our training, your team will be able to maintain such infrastructure with lesser overhead.

Easy configuration
The biggest argument for *not* using containers (or microservices) is complexity - but we have aleviated that by simplifying the setup. Akin to first networking hardware (such as routers), which used to be mandatory to be set up manually via command line, we have pioneered the UI for simple setup, as our forefathers did with first web-configurable routers and firewalls.

With our services, you get to choose between classical terminal/bash interface and our ContainerOS UI. The UI is simple multiple choice checklist for features you need. You can pick services from dropdown lists and specify the ports, IP addresses (and other technical data points) either by scanning the network and picking each one, or entering them manually in the UI text fields.

Contrary to most simplified UIs, though, you can configure just about anything in the ContainerOS UI without the need to access terminal. Hence both your sysadmin veterans and new staff will feel right at home, further saving you costs on highly experienced personnel.

Once you save the configuration, multiple tests are run to ensure 100% compatability between components to check integrity of the setup. If everything works well, you are just one step from easy deployment.

The stack

From the beginning, we strived for solution that is simple, but powerful. Preferably one that runs mostly on open-source code, as that is easier to track. We also donate to development of core technologies - we literally practice what we preach.

We were able to provide such a solution thanks to combination of Docker images with mandatory microservices, orchestrated by Kubernetes. This way we follow Open Container Initiative (OCI) and various best practices, using them as our foundation core.

Microservices
Microservices represent the layer running core infrastructure services, such as routers, networks, switches, VPN servers or firewalls.

Of course we realize that moving what was traditionally bare metal to virtualized container might seem like a proverbial leap of faith. Especially when it means trusting your business security to it. That´s why we work with best providers of microservice infrastructure, ones that are used by Fortune500 companies and other critical businesses. Also, we made the whole orchestration of containers robust to such extent that it only enhances what is already perfectly compliant to strict standards.

Dependencies
Our custom containers run flawless on Azure, Amazon Web Services and Google Cloud. Nevertheless, we are further developing them to be compliant with Digital Ocean, Rackspace, Heroku and other large cloud providers.

Why Kubernetes?
Kubernetes is among the most widely used container orchestration tools. It is also open-source, coming originally from Google Borg, hence why we trust it to be a solid choice.

Among the known benefits, we have chosen Kubernetes for:

- Automatic app/microservice health check. This means that when any service running in the container gets corrupted, it is restored from last working image, tested prior to deployment.
- Load balancing - routers and firewalls might expect spikes of traffic and are critical to stay operational. Thanks to superior load balancing of containers, rest assured that no spike is big enough to threaten safe operations.
- Redundancy means that even if single firewall or router fails (eg. due to hardware host fail), another one immediately takes place and yet another one is spun in the background for backup - hence you always have one working + one backup service.
- Safe updates - containers are rebuilt instead of patched when updating. Furthermore, updates are never scheduled to run at the same time - instead we update, test, rollback if needed and only if everything works, another sequential update runs. Only when all parts work in tandem, we deploy the whole updated stack.

How we configure and secure the containers

To boost the security, we configure each container cluster by following NIST guidelines, plus:

Separation of concerns
Each container can access only the needed resources and containers. No unnecessary shared folder on local network, or NAS drive floating about.

Granular privileges
As the whole cluster is defined by master, there is no risk of malicious intruders escalating privileges of a container. Also, each container is configured to access only specific files - not just groups or folders. Last but not least, passphrases and SSH (and other security) keys are again accessible only by those who need to use it.

Controlled access to resources
Cgroups and Namespaces are set upfront to limit the amount of CPU, memory, and network bandwith that each container can use.

Monitoring
Services are monitored not only from internal intrusion detection system, but also from outside the network with service running for this sole purpose (and no other privileges).

TLS/SSL everywhere
Secrets, passphrases and all transfers run over TLS/SSL of your choice. We even support Let´s Encrypt, but most frequent scenarios will run their own 3rd party certificates.

LDAP/Kerberos
Many two-factor authentication methods are available, whether it is Google Auth, PAM, JSON web tokens, USB keys such as Ubikey and Trezor or custom SMS push notifications. These are integrated into LDAP running in tandem with Kerberos.

Custom security practices
Even security-by-obscurity principles, port-knocking or fail2ban techniques might provide themselves viable for boosting your security. Thanks to automation, these again are not making life of your maintenance staff any harder - they just make it harder for potential malicious intruders.

Automatic destruction and rebuild
As with redundacy, should any node´s security be compromised, it automatically respuns itself from the last safe image and the security keys are changed. This way, even if anyone manages to take over a container, he is left with empty hands immediately upon intrusion detection.

How we deploy and orchestrate

Even the most robust solution would not be practical to use if it compromised easy deployment and control.

We can either make your custom Helm charts for specific use case, or pick from the growing library of published ones in the repository, where we add more on a weekly basis:

https://github.com/helm/charts

Easy deployment is further facilitated by your choice of who gets the privileges to deploy what, be it specific bare metal or virtual machines, connection over VPN and RDP and so forth. As the deployment service can be run as a middleware practically anywhere, the only real limit is the imagination and your specific business needs.

What if you have very custom scenario?
We get it. It all sounds too good to be true and as with every technology, even our solution might not be an instant fit.

When you no longer fit our configuration patterns or simply need different approach, we can tailor-made the solution for you. If it is very complex one, there is potential drawback in reducing the simplicity our product is all about. In such rare cases, we still think containerized infrastructure is the way to go, but if we discover we are not a good fit, we´re still partners and our consultants might at least help you find solution you need, even if it is not one provided by us.

How we train

In the begining, we have briefly talked about our solution being not dependent on our constant supervision.

Firm believers in DevOps principles, we provide training for your IT security and engineering staff. The beauty of our custom containerized infrastructure is the robustness on one hand and simple control and maintenance on the other.

We will train your staff and then supervise them for as long as you need, until you feel comfortable taking the proverbial training wheels off.

How we vet our own staff
Of course, for businesses just starting out, or in need of staff change, we provide ready-made teams.

Our staff is trained to know the containerized infrastructure inside out - so certificates such as CKA (Certified Kubernetes Administrator) are a must. Our engineers are also certified either in CompTIA Security+ or CISSP and other security industry standards. Linux fluency is maintained by mandatory LFCS or LFCE certifications as a bare minimum.

Why you might consider hiring us in the end
This has two benefits. First is that our specifically trained staff is much more effective in any tasks related to the setup and maintenace of secure containerized infrastructure.

The second benefit goes in hand with the first and means we can provide a contract with retainer payments that we keep at the price we have agreed upon, as we know our staff and how much time do they need. We will provide KPIs and SLAs that we stand by.

Bonus
At the same time, our DevOps principles promote redundancy and we are more than happy to help you hire new staff, train them with our internal staff and even help you set scheduled maintenance together with time and cost estimates your new staff will need. This way, you keep our professionals on board for only as long as you need. When your new team is ready, we move on, while remaining your support should anything go wrong or just confusing.

Conclusion

Containerized security infrastructure is a term that would sound like sci-fi just few years ago. We are not newbies trying out some new exciting concepts from Github forks. We are merely combining cutting edge tech with old-school dilligence and punctuality.

IT security is getting increasingly more important and complex, thanks to the rapid growth of the whole tech industry. But we believe it should not get increasingly expensive and stressful to maintain.

On the contrary - it should have certain standards and procedures, which, when followed, provide most reliable, yet replicable results.

This standard then furthers the spread of best practices followed not only across security engineers, but developers and even non-technical staff alike. So that tech is there for everyone to benefit from.

And this is what really DevOps has been for us right from the begining.