

Kubernetes and cloud native operations report 2021

Data from 1200 respondents on hybrid and multi-cloud operations, Kubernetes, VMs, bare metal, goals, benefits, challenges, operators, advanced usage, edge, and more.

Introduction

Canonical, the makers of Ubuntu, is a member of the Cloud Native Computing Foundation ([CNCF](#)) and provides commercial support for various technologies that are part of the CNCF ecosystem. As an active part of the community, we take the pulse, analyse and share insights on personal and commercial use of cloud native technologies, leveraging the Ubuntu user base as well as our proven experience working with open-source software and solving enterprise complexity. We contribute the data back to the community, along with our own analysis and the insights of industry experts in the form of commentary. The goal is to help improve the cloud native technologies to best address the needs of end users and their organisations.

Key takeaways

- This report includes responses to 50+ questions from **~1200 respondents and analysis from 7 industry experts**, representing Google, Amazon, the co-chair of the CNCF's operator framework, and thought leaders from across the space.
- Cloud Native is more than Kubernetes. While 45.6% of respondents report using Kubernetes in production, only 15.7% report using Kubernetes exclusively. **We explore usage of Kubernetes, bare metal, VMs, containers, and serverless applications.**
- Scale: we looked at the number of Kubernetes clusters (as well as best practices for cluster usage), and also at machines under management: 21.4% of respondents are managing more than 500 machines.
 - When analysed by Job Role:
 - **Devs** most commonly reported managing 1-5 machines (19.61% of developer respondents)
 - **Executives:** 20% reported managing 201-500 machines
 - **Management** had 2 answers tied for their top response: 13.8% reported managing 101-200, and another 13.8% manage 51-100.
 - **Ops:** 1-5 machines was their lowest response (3.8%), 10.55% reported managing 5000+.
- Change vs Stability: the report explores migration challenges, the demand for cloud native skills, hybrid cloud use cases, K8s and container tooling, best practices and more.

- Despite the unparalleled adoption rates of cloud native technologies during the past 5 years, both the ecosystem and its users have yet to cross the chasm of full enterprise adoption. 40% of respondents reuse traditional approaches from the VM world, (e.g. bash scripting), and 28% use Helm charts of varying maturity to manage software on K8s. To tackle enterprise software operation lifecycle management, operators might be the best solution, but the community has work to do to raise people’s awareness and confidence levels in them. 30% of respondents suggest that trying out operators is on their to-do list.
- 22.5% of respondents called out making coffee as the one thing they would like to do with K8s and currently can’t. Alexis Richardson called out the bold 11% who said “I can do everything I need using Kubernetes”, by describing this as “Kubris.”
- Off-beat statistic*: Panda-lovers are 50% more likely to be running 50+ Kubernetes clusters (the largest number of clusters people could have chosen in the survey).

About the survey

Note, the survey is still open — let your voice be heard & share your insights with the community: [Complete the survey here](#). Note: in leaving the survey open, the [raw data](#) may be different from the data used at the time of the report’s publication.

Industry expert bios



Michael Hausenblas
Solution Engineering Lead



Michael is a Solution Engineering Lead in the [AWS](#) open source observability service team. He covers Prometheus/OpenMetrics, Grafana, OpenTelemetry, OpenSearch, and Fluent Bit upstream and in managed services. Before Amazon, Michael worked at Red Hat, Mesosphere (now D2iQ), MapR (now part of HPE), and in two research institutions in Ireland and Austria.



Kelsey Hightower
Principal Developer Advocate



Kelsey is a Principal Developer Advocate at Google working on Google Cloud Platform. He has worn every hat possible throughout his career in tech and enjoys leadership roles focused on making things happen and shipping great software. He is a strong open source advocate with a focus on building great software as well as great communities around them. He is also an accomplished author and keynote speaker with a knack for demystifying complex topics, doing live demos and enabling others to succeed. When he is not writing code, you can catch him giving technical workshops covering everything from programming to system administration.

* The cloud native community is growing rapidly and delivering real business value — we believe that humour, passion, and technology go hand in hand. As such, there are some fun stats in this report that you won’t find anywhere else.



Tim Hockin
Principal Software Engineer



Tim is a Principal Software Engineer at Google, where he works on the Kubernetes, Google Kubernetes Engine (GKE), and Anthos. He has been working on Kubernetes since before it was announced, and mostly pays attention to topics like APIs, networking, storage, multi-cluster, nodes, resource isolation, and cluster sharing. Before Kubernetes, he worked on Google's Borg and Omega projects, and before that he enjoyed playing in the kernel and at the boundary between hardware and software in Google's production fleet.



Alexis Richardson
Founder and CEO



Alexis is the Founder and CEO of [Weaveworks](#), and the former chairman of the TOC for CNCF. Previously he was at Pivotal, as head of products for Spring, RabbitMQ, Redis, Apache Tomcat and vFabric. Alexis was responsible for resetting the product direction of Spring and transitioning the vFabric business from VMware. Alexis co-founded RabbitMQ, and was the CEO of the Rabbit company acquired by VMware in 2010. Rumours persist that he co-founded several other companies including Cohesive Networks, after a career as a prop trader in fixed income derivatives, and a misspent youth studying and teaching mathematical logic.



Karthikeyan Shanmugam
Digital Solution Architect



Karthikeyan (Karthik) is an experienced Solutions Architect professional with about 20+ years of experience in design & development of enterprise applications across Banking, Financial Services, Healthcare and Aviation domains. Currently engaged in technical consulting & providing solutions in the Application Transformation space. Karthik regularly publishes his technology point of view. His articles on emerging technologies (includes cloud, Docker, Kubernetes, microservices, cloud native development etc.) can be read on his [blog](#).



Ken Sipe
Senior Enterprise Architect



Ken Sipe is a Distributed Application Engineer working on the challenges around container orchestration. Ken is a co-chair of the Operator SDK and a committer on the CNCF sandbox projects KUDO and KUTTL where he's working to improve the Kubernetes experience. Ken is an internationally recognized speaker, receiving the Top Speaker and JavaOne Rockstar award in 2009, and continues to speak and lead projects on such topics as distributed application development, micro-service based architectures, web application security, and software engineering best practices.



James Strachan
Distinguished Engineer



James works on Jenkins X to help developers automate their CI/CD for cloud native applications and help them go faster. James also created the Groovy programming language and the Apache Camel integration framework.

Kubernetes and cloud native operations report 2021

Who is using Kubernetes and cloud native technologies?

1. Respondents by job role

I am best described as:

1162 out of 1166 people answered this question

23.7%	SRE/DevOps Engineer	275 responses
11.5%	Infrastructure Architect	134 responses
9.8%	Back-end developer	114 responses
8.7%	Full-Stack Developer	101 responses
5.5%	Academic/Teacher/Student	64 responses
4.6%	Security Engineer	53 responses
4.1%	Software Architect	48 responses
4.0%	Other	47 responses
3.4%	Consultant	40 responses
3.4%	DevOps Management	39 responses
3.1%	Engineering Manager	36 responses
2.9%	Front-End/Applications Developer	34 responses
2.8%	Data Scientist	33 responses
2.0%	Mobile Developer	23 responses
1.8%	CTO	21 responses
1.4%	Executive	16 responses
1.2%	Developer Advocate	14 responses
1.0%	Database Administrator	12 responses
1.0%	Machine Learning Specialist	12 responses
1.0%	Project Manager	12 responses
0.9%	Product Manager	10 responses
0.8%	Quality Assurance Engineer	9 responses
0.5%	Sales representative	6 responses
0.3%	CIO	4 responses
0.3%	Release Management	3 responses
0.2%	Marketing professional	2 responses

Across the board, industry experts felt this survey represented a fair distribution of the cloud native space today.



“Great distribution. Huge % of DevOps. Not a surprise for Kubernetes. Often, change agents are developers and many developers self-opt-in to the DevOps category. Sometimes you see organisations create “DevOps roles or DevOps teams” which act more like traditional Ops with a new name. Ideally, you’re hoping that people who list themselves as DevOps can wear both hats — they’ll code, read the logs and understand them, and have access to ops; all in one person”

—Ken Sipe, *Cloud Native Computing Foundation, Edward Jones*



“This data tells me two things: 1. That Kubernetes is predominantly for people who build platforms. They are different from developers who are interested in things like Ruby on Rails, gRPC, JS web frameworks, or iOS. If you group some of the categories here, it might even be safe to say that 50% of the people identify as being platform team members. 2. It also tells me that the tools these people are using are pushing them towards looking at Kubernetes. Why would a DBA be concerned with Kubernetes? Probably, because they’re starting to see some of the databases they’d manage, have a Kubernetes installation option. But DBA’s are only 1% here. And this looks accurate. So far, not a lot of database products say things like “here’s our community’s official Kubernetes operator”. So this is picking up, but it’s still early.”

—Kelsey Hightower, *Principal Developer, GCP Advocate*

Many of the questions later in this report have different answers, depending on who you ask — so we’ve grouped respondents by job role to see if we can gain extra insight. These groups can be broken down as follows:

Job roles segmented into “Role Categories”:

- Devs — 35.7%
 - BackEnd Developer
 - Full-Stack Developer
 - Software Architect
 - Front-End / Applications Developer
 - Mobile Developer
 - Data Scientist
 - Quality Assurance Engineer
 - Security Engineer
- Ops — 37.5%
 - SRE/DevOps Engineer
 - Infrastructure Architect
 - Release Management
 - Database Administrator
 - Machine Learning Specialist
- Management — 7.5%
 - DevOps Management
 - Engineering Manager
 - Project Manager

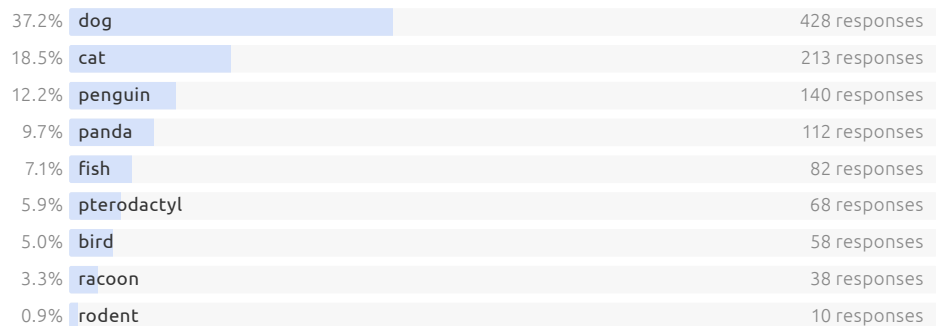
- Executive — 4.4%
 - CTO
 - Executive
 - CIO
 - Product Manager
- Academic — 5.5%
 - Academic/Teacher/Student
- Other — 9.3%
 - Developer Advocate
 - Consultant
 - Other
 - Sales Representative
 - Marketing Professional

Remember that the [raw data](#) for this report is publicly available. Feel free to analyze/segment/categorize the data as you'd like and share your findings with us!

2. Surrounded by Animals

You can only adopt one. Make your choice.

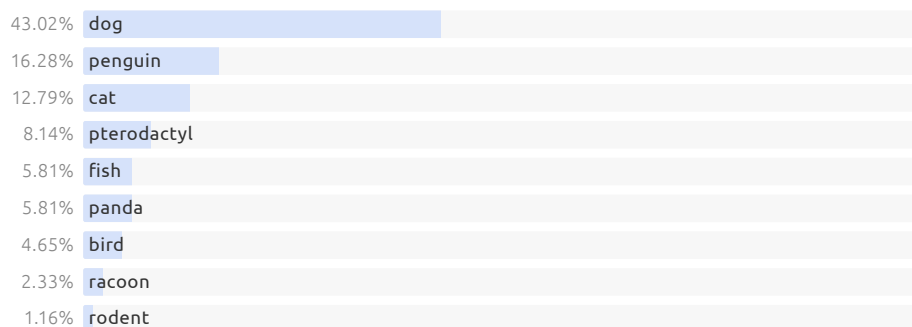
1149 out of 1166 people answered this question



We're not advocating pets over cattle here, but this "fun question" might have revealed the inner-workings of the people around you:

- All Job Role categories chose dogs, followed by cats, as the animal they'd most likely adopt, except one: management. Management chose dogs followed by penguins. Is there a reference to "herding cats" to be made here?

Management



- Panda-lovers are 50% more likely to be running 50+ Kubernetes clusters (the largest number of clusters people could have chosen in the survey).

3. Respondents by industry

My company is in this industry:

1165 out of 1166 people answered this survey

40.5%	Software/Technology	468 responses
9.3%	Financial Services	108 responses
8.1%	Education	94 responses
7.2%	Consulting	83 responses
6.4%	Telecommunications	74 responses
3.5%	Government	40 responses
3.4%	Healthcare	39 responses
2.7%	Professional Services	31 responses
1.9%	Manufacturing	22 responses
1.8%	Consumer	21 responses
1.8%	Scientific or technical services	21 responses
1.8%	Transportation and warehousing	21 responses
1.7%	Energy & Utilities	20 responses
1.6%	Retail	19 responses
1.4%	Media/Analyst	16 responses
1.1%	Arts, Entertainment	13 responses
1.0%	Non-profit	12 responses
0.6%	Military	7 responses
0.4%	Agriculture	5 responses
0.4%	Hotel and Food Service	5 responses
0.3%	Construction	4 responses
0.3%	Legal Services	4 responses
0.2%	Religious	2 responses
0.1%	Real estate, rental, leasing	1 response
2.2%	Other	26 responses

Throughout this report, the intention was to highlight areas where certain industries responded significantly differently to others. If you feel that your industry is underrepresented, the [survey is still open and you can share your responses here](#).



“I’m surprised Financial Services is #2 here. Often, this sector is highly conservative but, presumably, they’re learning about the new tech. The early days of Kubernetes had a lot of telco involved, good to see it’s still there. I’m keen to explore the financial angle in the answers to the other questions.”

–Ken Sipe, Cloud Native Computing Foundation, Edward Jones



“This confirms a lot of what we see in the real world. Even though a lot of people think financial services companies are all legacy — it turns out that even though they have all the risks and the security concerns, they’re typically the ones to adopt new technology sooner because they have margins to contend with.

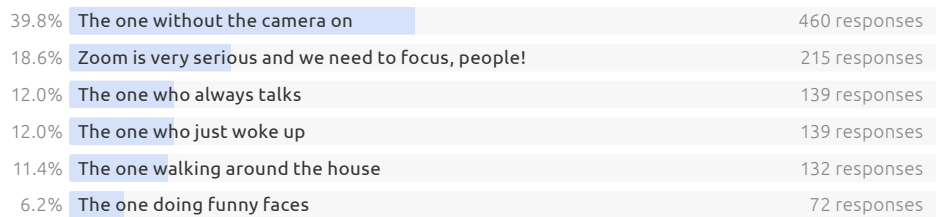
Telecommunications is another big one, too. Telcos are leveraging Kubernetes in their regional data centres for streaming devices and do things like scale to zero, or subscription-based dynamic rollouts of content. They have their custom networking stack and streaming protocols and use Kubernetes to integrate their data plane. Also, K8s has a much smaller footprint than virtualization platforms like OpenStack — this one is a bit heavy for the task. Kubernetes can run on bare metal and telcos tend to end up there for the performance, while also integrating their own control plane and start to leverage the automation available in K8s.”

—Kelsey Hightower, Principal Developer, GCP Advocate

4. Video conferences

What type of Zoomer (zoom-user) are you?

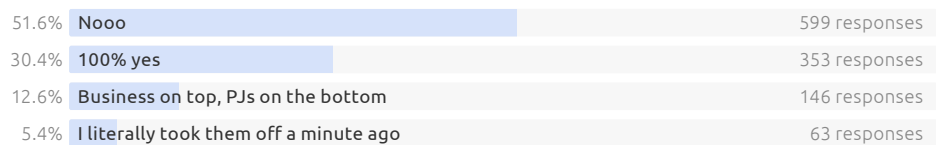
1157 out of 1166 answered this question



5. Work from home fashion

Hand on heart — are you wearing pyjamas right now?

1161 out of 1166 people answered this question



Although this question was intended to lighten the mood and help respondents make it through the 50+ question survey, there is a topic here worth addressing. It’s clear that the pandemic of 2019-2021 changed the way people work, driving up the numbers of remote workers. What does the future look like?

McKinsey and Company recently launched a report entitled, “[What’s next for remote work: An analysis of 2,000 tasks, 800 jobs, and nine countries](#)”. It’s an interesting read — worth opening in a separate tab.

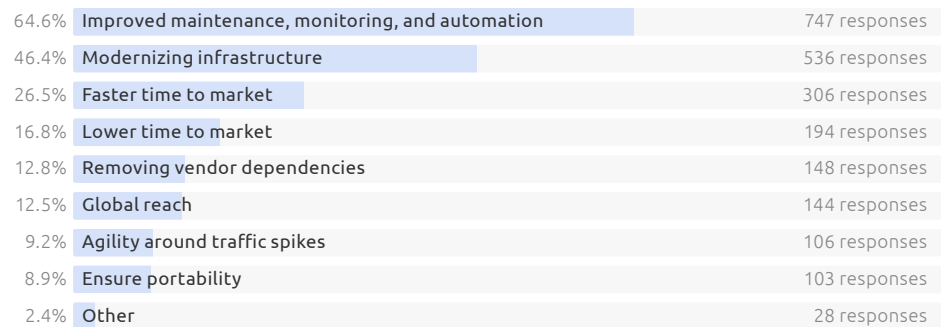
Interesting note: There’s a good chance that your boss is enjoying WFH almost as much as you are. 31.25% of executives admitted that they’re either wearing pyjamas or at least PJ bottoms during video calls.

Cloud native: goals, benefits, and estate size

6. Which technology goals are most important?

Which goals are most important to you and your team? Choose 2.

1156 out of 1166 people answered this question



Before looking at technologies or practices, it's best to understand people's goals — so that's what the report starts from.



"This looks to be super accurate. A lot of people think organisations are moving to Kubernetes because of scale, or because they want to be a hyperscaler, or have the same traffic levels as Twitter. That's not necessarily the case for the majority of organisations. A lot of people like the fact that many decisions are just built into K8s, such as logging, monitoring and load balancing. People tend to forget how complex things were, just to build an app without all that automation. If you were on a public cloud you could have used some of the native integrations and tools. But if you were on-prem, that was not a given — you had to go and glue the solution together yourself. With Kubernetes, you almost collapsed 25 different tools into one. This is what people mean when they say 'modern infrastructure' — they're not literally talking about doing something that has never been done before. They're talking about the things that have been in production for the past 10 or 15 years. Kubernetes is today's checkpoint on all the 'modern patterns'."

—Kelsey Hightower, Principal Developer, GCP Advocate



"One of the things Kubernetes has really helped with is standardising infrastructure. It used to be that everyone's on-premise was completely different — different control planes, tools, infrastructure, everything. What we all want is to improve the maintenance and monitoring of all of our software. And we want that to just work. These first two answers match my thinking — we want a modern standard way of running software that works in every way."

—James Strachan, CloudBees



“Maintenance, monitoring and automation is a big category — that’s a lot of things coupled together. I’d propose to separate automation from monitoring and maintenance. Maintenance could mean versioning of Kubernetes (and deciding if you want to keep up with the Kubernetes release cycle), but it could also mean maintaining your own tooling and/or infrastructure. I’d expect security-sensitive organisations to want to control more things — to build out their own infrastructure perhaps. There is room to explore this more.”

–Ken Sipe, *Cloud Native Computing Foundation, Edward Jones*



“I think that’s one of your flagship results. We see automation being a big theme; it’s about lowering operating costs and it also ties into faster time to market, which is definitely a good business driver. Another theme that we see has to do with flexibility, as expressed from the number of people looking for portability and removing vendor dependencies, which could be linked to modernising infrastructure.”

–Alexis Richardson, *Founder and CEO, Weaveworks*



“This result correlates with the survey’s sample population, the majority being SRE’s. Improved maintenance, monitoring and automation are most important to this group of people.*”

–Karthikeyan Shanmugam, *Digital Solution Architect, HCL*

* Note: Karthik’s comment was bang on. When we dug into the data, we found that 77.8% of SREs/DevOps selected improved maintenance, monitoring and automation as one of their two choices for this question.

7. What are the top benefits of cloud native technologies for businesses?

Kubernetes and cloud native technologies provide a platform for businesses to innovate and achieve their goals. Out of the benefits they bring, which are the most important for businesses? We asked respondents to grade the following benefits on a scale of 1 to 5, with 1 being the least and 5 the most important:

- Cutting-edge technology
- Developer productivity
- Elasticity and agility
- Global reach
- Open-source software
- Portability
- Reduced OpEx and CapEx
- Resource optimisation
- Simpler operations

No particular definition was given for the different benefits in the survey, users were allowed to interpret them as they saw fit. The benefits are listed from most important to least, according to their average grades in the survey:

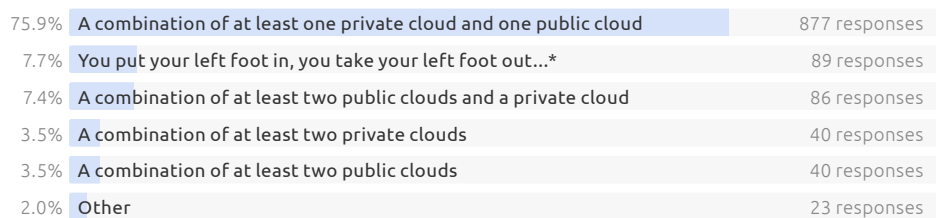
- Elasticity and agility — Average 4.2
- Resource optimisation & developer productivity — Average 4.1
- Faster time-to-market & simpler operations — Average 4
- Portability & open-source software — Average 3.9
- Cutting-edge technology & global reach — Average 3.8
- Reduced OpEx and CapEx — Average 3.6

In general, all users, regardless of their job group, ranked the cloud native benefits similarly, with a standard deviation of approximately 0.15. The only data point that differs is reduced Opex and CapEX (3.8 where the average was 3.6) and faster time to market (4.3 where the average was 4) being more important to Executives than the rest of the job groups.

8. What is the definition of “hybrid cloud”

How do you define a “hybrid cloud”

1155 out of 1166 people answered this question



*You put your left foot in, you take your left foot out, you put your left foot in, and you shake it all about.

It’s really interesting that people are confused about the term “hybrid cloud”. While 75.9% share a similar definition, there’s a big enough group who have their own views. End result: confusion. When we add the term “multi-cloud” to the mix, we assume that the confusion will only grow. Thankfully, [Tytus Kurek](#) has clearly [defined hybrid cloud and multi-cloud](#).

And now, for today’s moment of zen, we give you Ken Sipe:



“Reality is what the majority of people perceive it to be.”

—Ken Sipe, Cloud Native Computing Foundation, Edward Jones

9. What are your hybrid cloud or multi-cloud use cases?

In conversations with industry experts, Kelsey Hightower mentioned something interesting. He suggested that if we grouped “accelerate development”, “automate DevOps” and “disaster recovery”, then you could say that “often, the first use case for multi-cloud is to handle the things we can’t do: back up our own stuff and get super quick capacity to unblock Dev in case of a disaster.”

We mainly use hybrid or multi-cloud for/to:

1154 out of 1166 people answered this question

22.2%	We don't use hybrid or multi-cloud	256 responses
20.7%	Accelerate development, automate DevOps	239 responses
13.3%	Expand cloud backup options to cut costs	153 responses
12.6%	Disaster recovery	145 responses
6.2%	Cluster mission-critical databases	71 responses
5.5%	Move an application	64 responses
5.3%	Cloud bursting	61 responses
4.2%	Switch between public cloud providers in a flash	48 responses
4.0%	On and off-ramp data	46 responses
2.4%	Monitor and predict usage costs	28 responses
3.7%	Other	43 responses

Overall, 77.8% of respondents are using hybrid or multi-clouds. As this technology continues to progress through the technology adoption curve, new challenges are going to arise — particularly around scale. We're already seeing scale issues in the bare metal, VM, and Kubernetes spaces, often referred to as "sprawl". The movement towards "model driven operations" is an example of people aiming to solve these scale issues.

Relevant Quotes:



"There are different use cases and motivations for hybrid and multi-cloud. For example, hybrid is very often used for app migration or cloud bursting, whereas multi-cloud can be more related to cost consolidation or even acquisitions — resulting in different workloads on different cloud providers."

—Michael Hausenblas, Solution Engineering Lead, AWS



"People dip their toe in before they adopt. I see bursting into the cloud based on scale as a reasonably advanced technique, so I'm not surprised that it's not done by a majority of people yet."

—Ken Sipe, Cloud Native Computing Foundation, Edward Jones



"Multi-cloud is an interesting market driver, especially for regulated companies. I was recently talking with a banking customer who said they were interested in multi-cloud and hybrid cloud because the regulation dictates they should avoid having a high concentration of risk on a single cloud."

—Alexis Richardson, Founder and CEO, Weaveworks



“22% of people say “I’m not using hybrid or multi-cloud”. This looks a little too high to me. I think people might be saying, “we’re not running things under our domain in places we do not own”, but they often forget that their organisation probably uses Salesforce or email services as part of day to day operations and these definitely run in somebody else’s cloud. If they include SAAS in their multi-cloud definition, I’d expect this number to be much lower.”

–Kelsey Hightower, Principal Developer, GCP Advocate

10. Size is important: How many machines and clusters are people running?

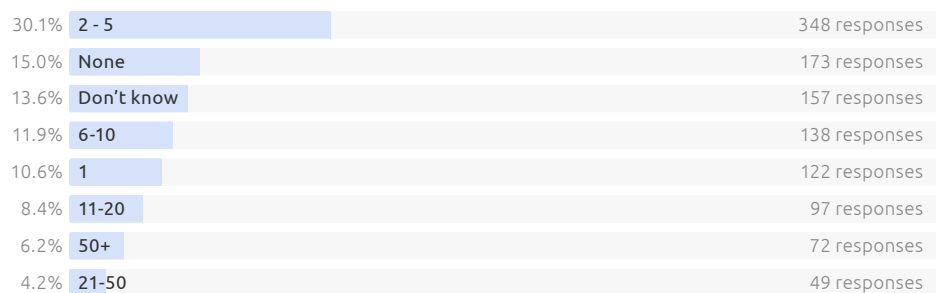
On average, how many machines are in your fleet (incl VM, bare metal, etc)?

1158 out of 1166 people answered this question



If you use Kubernetes, how many production clusters do you have?

1156 out of 1166 people answered this question



Whether you’re in the gym, at the kid park, or out in town, everyone is talking numbers. How many machines are you running? How many clusters? How big are the nodes? What type of nodes (CPU, GPU, etc)?

21.4% of respondents are managing more than 500 machines. It’s worth examining these responses broken down by job role. Devs most commonly responded 1-5 (19.61%), while nearly all other job role categories reported managing significantly more machines than the devs. Their top answers were as follows:

- Executives: 20% reported managing 201-500 machines
- Management had 2 answers tied for their top response: 13.8% reported managing 101-200, and another 13.8% manage 51-100.
- Ops: 1-5 was their lowest response (3.8%), with their top 4 being 51-100 (14.7%), 201-500 (12.9%), 101-200 (12.2%), and 5000+ (10.55%).

And here's what our industry experts had to say about all this:



"This speaks to what we saw previously: people are using Kubernetes for things like better config management or like a better automation tool. They're not waiting until they get to a 200 node scale problem where current tools fall apart. My guess is that 20 to 50 node clusters is probably the sweet spot for most people. For some organisations, it makes sense to have a lot of small clusters, one for every team or one for every project, for example. But, in order for that analysis to be more accurate, I'd need to know how many K8s clusters they have and then how big are these clusters."

—Kelsey Hightower, Principal Developer, GCP Advocate



"It's interesting to think about use cases of clusters. What size do they have for given use cases? A cluster dedicated to analytics often cycles between times where it's heavily utilised and others where it's not. Web apps often tend to have times with gaps in resources because they don't have high density, so they're ready to scale — but in off-peak or off-season times, they use less. In using the cloud, have organisations decoupled capacity and the ability to scale in a way that allows them to save money? I'd like to ask them what they're using to manage this."

—Ken Sipe, Cloud Native Computing Foundation, Edward Jones



"I've spent a lot of time recently talking to customers about their use and experience with single and multiple clusters. I can probably name 25 different answers to the question "Why do I need multiple clusters?". Some of them are obvious, like "I want to put a cluster in Europe and a cluster in the US and I want to load-balance between them". There are others that make some sense, like "I have a really high-value application, and I want to just give it extra safety so I put it in its own cluster". Some are kind of FUD: "I don't trust the way that Kubernetes does isolation between applications. So I'm going to put it in another cluster and put all my apps in their own clusters". Others, I think, can potentially be disastrous at scale: "I'm going to give every team their own cluster and let them do whatever they want".

The question, I think, is really interesting; essentially, it is: 'how do you think about your clusters as an organisation? Are they a set of pets? Or are they actually cattle in a herd that you want to manage together?' SIG multi-cluster has been trying to figure out models that work for people, with moderate success. There are also a few companies, like Rancher and Canonical, looking at this problem from a different angle. Something like a one or two machine installation in the back office of every store is very different from our customers at Google asking us to put 1000 nodes on a single cluster."

—Tim Hockin, Principal Software Engineer, GCP



“I still see people treating clusters more like pets rather than cattle. They deploy clusters with hundreds of nodes and keep adding more nodes to them. This can lead to single clusters of extreme sizes. So how do you do isolation or multi-tenancy in that case? What is your upgrade strategy? Do you require zero downtime? Have you considered the blast radius? Instead, if they went with a more cell-oriented architecture, where every team has its own cluster, or even multiple ones for dev, test and production, the separation of concern would probably simplify the operations. Overall, this is a maturity indicator.”

–Michael Hausenblas, *Solution Engineering Lead, AWS*



“If we take out the “Don’t know’s”, 60% of people have more than one cluster. To those people, I would ask: do you have policies that you want to keep the same across those clusters? How do you enact them? Do you use your multiple clusters for load balancing or geo-distribution? Or do you just have one cluster for dev and another one for prod? Are you afraid of cluster upgrades? Afraid of the blast radius? If you mess up the cluster upgrade, does that mean your entire business is taken down? How do you defend against that? Those, to me, are the interesting questions around multi-cluster.”

–Tim Hockin, *Principal Software Engineer, GCP*



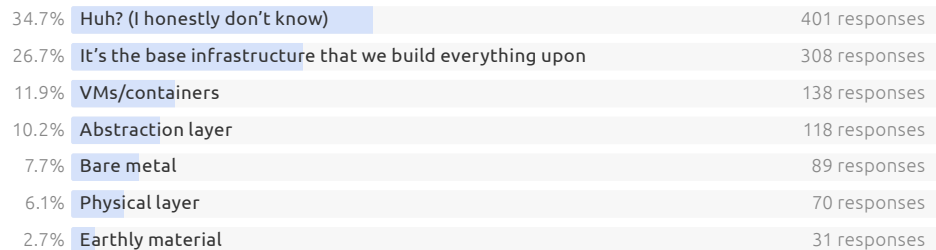
“This is in line with my experience of talking to the community. I see people choosing to split their Kubernetes clusters between at least two zones, to give them a little bit of high availability, thinking about failure domains and blast radius — that’s just smart. Now, 2-5 clusters might be about a multi-region setup or compartmentalisation. An example of the latter can be having a dedicated cluster for databases, one for all the web apps and the rest of the clusters for the more sensitive apps and/or data. People probably still prefer to isolate these at the cluster level with their own set of dedicated machines. 10% of the people are doing 6-10 clusters. That could mean that there’s two for dev, two for the UAT environment, and then two for prod that gives you the total of six. That 50+ range is where we might see one cluster per team or per business unit. Here’s where multi-cluster tooling is necessary to deal with the challenge of cluster management at scale. A single cluster might be for people that still are at the planning/experimenting phase, the first step before going into multi-cluster.”

–Kelsey Hightower, *Principal Developer, GCP Advocate*

11. Defining “substrate”

In the infrastructure space, what does the term “substrate” mean to you?

1155 of 1166 people answered this question



Substrate is a word that we hear a LOT in internal conversations. In a multi-cloud or hybrid cloud world, sometimes people are building on bare metal, sometimes VMs, sometimes Kubernetes, sometimes VMs on Kubernetes — so it helps to have a term to handle all of them at once. We like “substrate”. What do you use?



“I tend to associate the word “substrate” with networking. That, to me, is the most fundamental thing as it connects storage, traffic, enables the communication between all the things running in a cluster. Going above networking and towards Kubernetes, we often refer to “platforms”, like the compute layer is a platform or the data platform.”

—Kelsey Hightower, Principal Developer, GCP Advocate



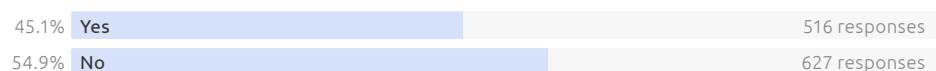
“While we’re defining terms, it would be interesting for DevOps to define “control plane”. This tends to be a vendor-defined term, and the challenge we have is that Kubernetes is the control plane for the most part (in K8s-only environments), but now that “operators” are opening up with “Charmed Operators” across substrates, charmed operators act as a “control plane extension” for k8s, containers, VMs, bare metal, etc.”

—Ken Sipe, Cloud Native Computing Foundation, Edward Jones

12. Are companies contributing code back to open source projects they use?

Is your company contributing code back to the open source projects that you use?

1143 out of 1166 people answered this question



The open source community is built upon contributions from people all over the world. 45% is a strong strong number — and if we dig deeper, 55% of company representatives from the IT industry and 50% from telcos and retail say theirs is an open source contributor. Well done!

Container and Kubernetes usage

13. Applications platforms

Where do you run applications in your organisation?
Choose the most accurate statement.

1156 out of 1166 people answered this question

29.9%	On a mix of bare metal, VMs and Kubernetes	346 responses
15.7%	All our applications are on Kubernetes	182 responses
15.3%	Mostly on VMs, planning a full migration to Kubernetes	177 responses
13.1%	On VMs, evaluating Kubernetes for development	152 responses
5.8%	Don't know	67 responses
3.6%	Other	42 responses

Is everybody swimming deep into the K8s ocean? Or are we still just dipping our toes in the water? Almost 30% of our respondents run applications on multiple environments and only 15.7% are all in on Kubernetes. Every year the usage of Kubernetes as the de-facto platform to run application estates increases ever so slightly. While we're excited about this — as it means that the industry is slowly but steadily standardising the way people deal with infrastructure — we're also 100% supportive of the idea that the best infrastructure for your applications is the infra combination that makes the most sense to YOU. As Mark Shuttleworth said in the KubeCon Operator Day keynote this year: "Kubernetes? Yes. VMs? Yes. Bare metal? Yes. Public clouds or private clouds? Yes."

No matter your substrate, once those decisions are made, there is a shift in focus to the next big challenge: automation of operations.



"Only 15% fully using Kubernetes. So I think this clearly shows we've got a long way to go before we've properly modernised the infrastructure, until "the rest" is just part of the Kubernetes backplane."

—James Strachan, CloudBees



"This is probably super spot on. I don't think anyone is all-in on Kubernetes. What we're seeing is that organisations coming from traditional approaches are moving things to SAAS. And then what's left can probably go on Kubernetes. Managed services also fall in that SAAS category. So I think you have to dig into that 15% (All our applications run on Kubernetes). Why are you running everything on Kubernetes? And how many of your things went to managed services, instead of being ported from VMs to K8s? That's the reality right there, a very mature set of responses. Kubernetes is great for lots of workloads, and it keeps getting better every year. But the truth is, (e.g.) for some interactive machine learning jobs, you just want the biggest piece of bare metal you can get with no additional layers, you don't really care about centralised logging, you just want speed. So it makes a lot of sense that people are mixing and matching. The idea that everyone's using Kubernetes is probably not 100% accurate, but I think it is true that everyone is experimenting, that there is a small Kubernetes cluster somewhere in every org even if it's only a POC."

—Kelsey Hightower, Principal Developer, GCP Advocate

14. Kubernetes will bring world peace. Change our minds



An astounding 74% of the surveyed sample could relate to situations like the one described in this meme (based on the [famous Dilbert comic strip](#)). Although Kubernetes has evolved a lot during the past years, the rest of the survey clearly depicts cases where it has yet to evolve further to address a wider range of business requirements.

15. What cloud native use cases are you working on?

Which of these cloud-native use cases are you working on now?

1156 out of 1166 people answered this question (with multiple choice)

36.3%	Deploying or managing Kubernetes-as-a-service	420 responses
34.0%	Re-architecting proprietary solution into microservices	393 responses
26.7%	Moving to an open-source solution	309 responses
25.7%	Orchestrating workloads across a multi-cloud setting	297 responses
25.3%	Managing or enabling a hybrid-cloud setup	284 responses
24.6%	Using best-of breed cloud-native tools	284 responses
20.0%	Deploying business solutions in different geographies	231 responses
13.6%	None of the above	157 responses

In this question, we let respondents select as many answers as appropriate to their situation. The top 3 answers were: deploying or managing Kubernetes-as-a-Service (KaaS), re-architecting a proprietary solution into microservices and moving the business from a closed-source to an open source solution.

From a job role category point of view, Ops are mostly (48%) working on KaaS deployment whereas Devs are focusing (35%) on re-architecting applications.

Overall, these are mostly “easy” use cases, which tells us again that the majority is still early in the cloud native journey. Looks like Ken Sipe also agrees with us, while adding further considerations:



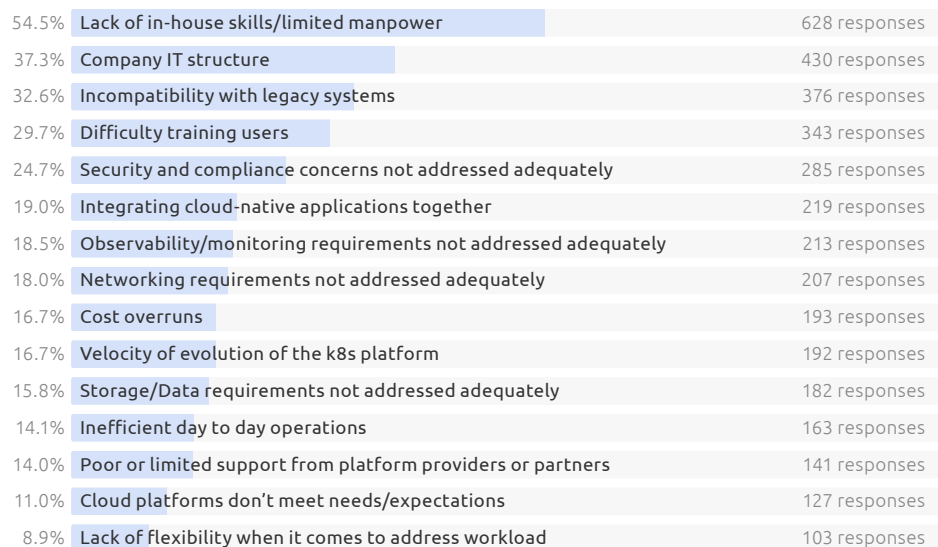
“The top two answers are consistent with my view of the industry. Most people are only at the beginning. The majority are deploying and managing Kubernetes as a service — they’re getting AKS, EKS, or something similar and they’re re-architecting apps. I’m seeing both of these. The challenge is: when you have this at large scale, monitoring bandwidth becomes one of your most important activities. You notice that one application with hundreds of instances becomes too “chatty”, or demands too much bandwidth. This can actually saturate the trunk, so much so that it affects your other services, and solving that becomes your next challenge. You’ll need to move data to the cloud, which a lot of organisations, e.g. in the financial industry, are challenged to do, or you have to increase bandwidth or segment, or implement a combination of the two.”

–Ken Sipe, Cloud Native Computing Foundation, Edward Jones

What are the top challenges Kubernetes brings to businesses?

What are your biggest challenges when migrating to/using Kubernetes and containers? Select all that apply.

1153 out of 1166 people answered this question (with multiple choice)



Many organisations are using or experimenting with Kubernetes, but these results help illustrate that usage has not permeated completely. Nearly 55% of respondents believe they lack Kubernetes skills and in-house expertise. This leads to challenges tuning the platform for optimal security, observability, networking and storage configurations, challenges training users (29.7%) and resistance due to company IT culture (37%) or monolithic architectures by 33%. Our experts mostly empathise with the users and provide a positive perspective and solutions to the problem:



“People are often struggling to start with Kubernetes. What I often tell them is that, in truth, they already know 80% of what’s going on in Kubernetes, if they come from the Linux community. Their cluster probably runs on Ubuntu, their apps are already written to run on Linux, the apps are going to make the same system calls, look for the same config files and run on the same kernel as before Kubernetes. If they start from the base principles, they can have their apps running on a newer version of Ubuntu and that will bring them at 90%. It’s really all about using a different tool to automate. They may lack the confidence that comes with the transition to a different stack, but they should rest assured that the basics are the same.”

—Kelsey Hightower, *Principal Developer, GCP Advocate*



“Everyone is still struggling with regards to in-house skills around Kubernetes. It really is a matter of buy versus build though. There are companies keen to invest in building up their skillset and engineering power so that they can run the entire stack and even hire evangelists that preach and lead with good practices. Other businesses say “We’re not in the business of Kubernetes, we just want the benefits”. They are happy to outsource the skills to consultants and professional services, only keeping a limited number of in-house experts to ensure smooth operations. They are also the prominent users of the DIY PAAS solutions that completely shield the low-level details. In the end, developer teams don’t really see a difference. Where they previously had a tool that provisioned their VM, now they have something similar that creates a pod in a namespace.”

—Michael Hausenblas, *Solution Engineering Lead, AWS*



“I find the lack of skills answer particularly interesting. I believe people underestimate how big of a technology leap there’s been in the last five years: Kubernetes, cloud, CICD, DevOps, GitOps, ChatOps. So much has changed. And it’s really hard for lots of people to keep doing their job, but also learn all this new stuff. Lack of skills and training manpower is a huge issue right now. And I think the only way to really learn is by doing; pick a path and try the new tech.”

—James Strachan, *CloudBees*

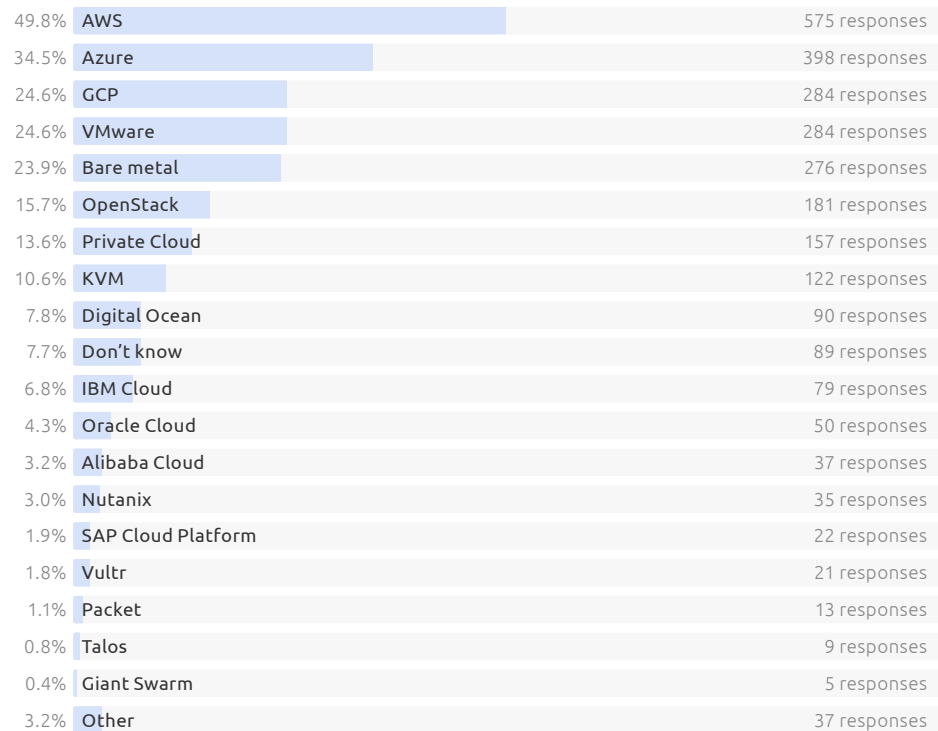
If any of these concerns resonate with you — Canonical can help.

- Get the in-depth analysis of Kubernetes challenges and solutions for the enterprise in [Enterprise Kubernetes: A buyer’s guide](#)
- Learn about Canonical’s Kubernetes managed services, deployments, and multi-cloud operations support here: <https://ubuntu.com/kubernetes>

17. Where do you run Kubernetes?

In which environments do you run Kubernetes clusters? Select all that apply.

1155 out of 1169 people answered this question (with multiple choice)



Respondents were able to choose multiple answers to this question — based on the multiple environments they're running. If we dissect the data, it looks like the first wave is made up of public clouds, who have been a steady wave of innovation for the IT industry over the past decade. That is also true for Kubernetes and the cloud native space. The second wave of responses are all on-prem solutions such as VMware, Openstack and bare metal. Following them, a long tail of various answers with smaller percentages shows the diversity of environments, something that speaks to the universality of Kubernetes as a platform.



"I think this really speaks to the original promise that Kubernetes would be a vendor-neutral solution to the cloud space. It's not a Google-only technology. It's something a lot of people contribute to. What's really interesting here is the bare metal component. Do

you really need a lot of VM provisioning superpowers? Probably not. You're probably going to be able to get more performance if, for certain types of apps, such as Kubeflow for machine learning or some high-performance web applications, you get rid of some of the VM management layers and just give the whole raw compute layer to Kubernetes. Kubernetes has APIs for attaching storage volumes or managing IP ranges, so why add the hypervisor overhead?"

—Kelsey Hightower, Principal Developer, GCP Advocate

18. Kubernetes versions and upgrades: too fast or too slow?

What is the oldest version of Kubernetes that you have running on a production cluster?

980 out of 1166 people answered this question (with multiple choice)



Kubernetes releases happen, as of 2020, three times a year. The community is adamant about fixing issues and improving the technology quickly. But is that same pace being adopted by industries? Are they able to keep up and benefit from the latest improvements? The survey showed that this is an area for improvement with 26% of respondents at least 4 versions behind the latest stable. We will let the actual maintainers and contributors speak their mind:



“People are stuck between a rock and a hard place on this one. On one hand, they say it’s hard to keep up with releases — enterprises in particular struggle to absorb a new release every quarter. On the other hand, slowing down a release doesn’t mean the development will slow down too, which means more risk will be accumulated in every release. Most providers like Google or Amazon pick up the release many months after the initial release date. This means that if I fix a critical bug, it can take between nine to fifteen months for the end customers to receive it. As developers, it’s really difficult to get good feedback this way. We do a lot of testing, we have our own CI, but I usually only have a few critical users try something out, bulk data is hard to find.”

—Tim Hockin, Principal Software Engineer, GCP



“This tells me that 26% of the people may have rolled their own cluster, maybe they’re not quite ready to use a distribution. And they’re just using a homegrown or community-based deployment tool. For the 1.17 to 1.19 version these are either people that are really good at managing K8s on their own or using a managed service or a cloud provider. The data is also aligned with the mentality seen in the Linux distribution world, where we have things like Ubuntu LTS. People don’t just upgrade to the newest version of Kubernetes just because it got released. They will kick the tires to see what it brings, way before moving to it in production.”

—Kelsey Hightower, Principal Developer, GCP Advocate



“Let’s put that in perspective: 1.17 is over a year ago. And 1.10 is 2,5 years ago. So people are using software that has bugs that have been fixed for 2,5 years. I’m sure they’re unhappy about it. All I can say to them is to upgrade to a newer version because it’s fixed. As a community, we cannot backport changes to 1.10 because we just don’t have the capacity for it. This is where vendors like Canonical or Red Hat make their hay, with extended support contracts. If somebody is willing to pay for that, and someone’s willing to fund the engineering to keep backporting those changes, more power to them.”

–Tim Hockin, Principal Software Engineer, GCP



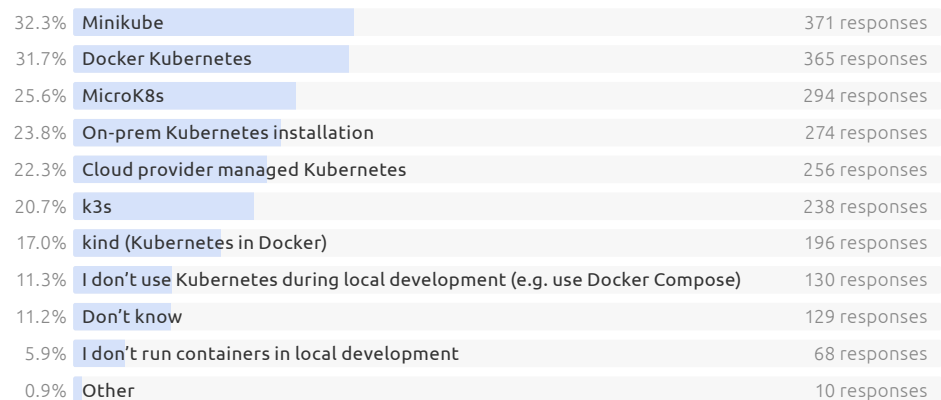
“Again, this comes back to pets versus cattle on the cluster level. If you do have a big — hundreds or thousands of nodes — cluster, then you will probably be very conservative with upgrading. On the other hand, if you treat your clusters immutably, like cattle, then a few of them might be on an old version, but I would expect the majority to be on the latest versions.”

–Michael Hausenblas, Solution Engineering Lead, AWS

19. Kubernetes in local development

What Kubernetes environment(s) do you target during local container development? Select all that apply.

1150 out of 1166 people answered this question (with multiple choice)



Lightweight Kubernetes distributions lower the barrier of entry to Kubernetes and are being used more and more by developers on their local workstations, as the first step of their development workflows. Depending on how one would look at this, these can increase efficiency and quality, by bringing developers closer to real production environments and standardising their workflows, or, if used thoughtlessly, can make the same workflows more complicated and error-prone. In any case, distributions like MicroK8s and k3s, following the path that Minikube started, are drawing increasing attention and now span use cases even beyond local development.



“I’m a big advocate of using Kubernetes for development. That said, I think people should use the same kind of cluster for local development as the one they have in production, just use a different namespace. There are tools out there to help build container images on Kubernetes, try them out and experiment. Why learn different things like Docker or docker-compose if, in the end, you want to run containers on Kubernetes? Have the same infrastructure for dev, staging and production in K8s and use that for everything. That will encourage developers to learn what production is really like and allow them to uncover issues early in dev more easily.”

–James Strachan, CloudBees



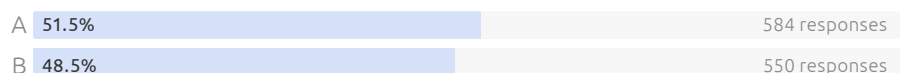
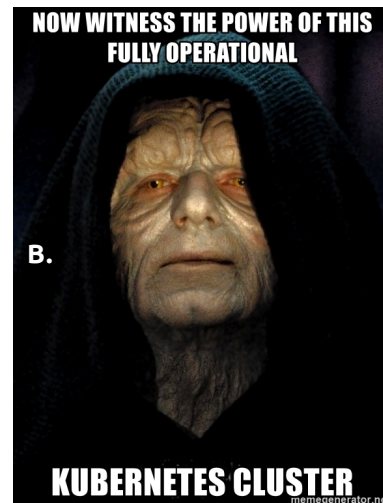
“I personally don’t use Kubernetes for local development. My take is you’re never going to be close to production on your laptop, there’s just so many things that will never be the same. This speaks to the fact that there’s no good way to share things in a remote cluster and there are no great packaging tools, so people can’t install different versions of the apps, so this is why there’s demand for local K8s. I’m surprised there’s no one just saying “I’m just using Docker, just Docker, no Kubernetes”. I think people are painting themselves into a corner here. Using Kubernetes on their laptop, trying to have a similar workflow as the production deployment. But the truth is, the secret is going to be different. The network policies are going to be different. And so things will work on the laptop but might not work in production. I expect these numbers to change, but again, it’s going to depend on the evolution of developer tools.”

–Kelsey Hightower, Principal Developer, GCP Advocate

20. What is the best Kubernetes meme?

Which meme?

1134 out of 1166 people answered this question



At this point we just wanted to decompress by nerding out a bit. We picked our favorite two Kubernetes memes and asked our respondents their favorite one. “Mr. kubeconfig” won by a very close margin with 51.5%. Not that this is saying anything, but executives were the only job group that went with the “K8s Sith Lord” as their top choice with 57%.

Mini-competition: Please, tweet your favorite K8s meme and tag @ubuntu.* This is super important to us. You might even win a Charming t-shirt!

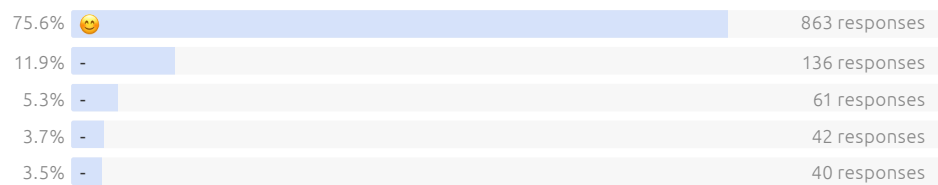
* By tweeting you agree to Canonical’s [Privacy Policy](#), [Privacy Notice](#), and the competition’s [Terms and Conditions](#).

Configuration Management

21. Yet Another Markdown Language? :-)

Which of the following is the best way to make your life easier with YAML?

1142 out of 1166 people answered this question



This question was meant to be slightly tongue-in-cheek, and since we couldn't find great answers, we opted to smile instead. Whether you hate it or love it, yet another markdown language (yaml) is here to stay. The cloud native community is working meticulously to abstract complexity from the different solutions, including automating many of the low-level configuration steps that have people screaming “YAML!” in their sleep. Brighter days are ahead!



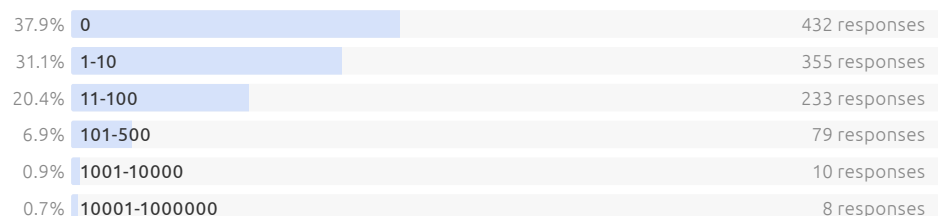
“I can certainly relate to this, .yaml files are tough to maintain and open the door to a lot of human errors that can even mess up entire deployments.”

–Karthikeyan Shanmugam, Digital Solution Architect, HCL

22. How many Helm charts has your team used/modified in the last 90 days

How many Helm charts has your team used/modified in the last 90 days?

1141 out of 1166 people answered this question





“Helm is a package manager, it stores versioned Helm charts in a repository. It also does a form of configuration management that allows you to templatised yaml. Some developers are put off because of the templating side, but they don’t realize that’s completely optional. You can just use Helm as a way of versioning tarballs of yaml which on its own is really useful. Helm needs to be standardised so that everyone just gets on with it.”

–James Strachan, *CloudBees*



“Interesting. People are using it but in small amounts. I think zero is the most interesting part of this. It’s either that people aren’t using it yet, or that they’re intentionally planning not to. I’d like to know which.”

–Ken Sipe, *Cloud Native Computing Foundation, Edward Jones*



“This aligns with my own anecdotal data of helm being attractive at small scale and counter-productive at a larger scale.”

–Tim Hockin, *Principal Software Engineer, GCP*



“This whole thing with Helm resembles the days of configuration management, with Chef, Ansible, Puppet etc. Helm is really great to get people started, inexperienced users can use somebody else’s blueprint and just run with it. When the need comes to start tweaking things, that’s where it gets complicated. More major changes and you get to a point where you’re essentially on your own and have to keep track of what the community is doing and try to align. The “1-10” response might be people using official third party Helm charts, like when I go on the Redis website and it says “the best way to try our product on Kubernetes is to use this Helm chart”. Fine. When I see the “100 to 500” response, that to me is people using Helm for their own, custom apps. And that percentage is still fairly low. The zero Helm charts group is a mix of people at the start of their journey with K8s or just the fact that people won’t trust unofficial, non-bulletproof Helm charts for third-party software in production, which makes sense.”

–Kelsey Hightower, *Principal Developer, GCP Advocate*

These expert opinions are quite interesting and seem to converge. We’re also hearing from enterprises interested in moving from configuration management to application management and model driven operations to handle Day Zero through Day 1000. Management: [Applications over configurations](#).

23. What is the best way to manage software on Kubernetes?

What is your preferred method for operating, upgrading, and maintaining software on Kubernetes?

1139 out of 1166 people answered this question



28% of respondents use Helm as the top option to manage software on K8s. Helm's popularity stems from the fact it made it easy to get started with container deployments, despite the fact that it cannot handle operations in the long term. Once again, our experts had very insightful comments to make about this particular topic:



"If you combine bash scripts and config management, that's 40%. That's really telling me that the operations folks are bringing their existing tooling over to the K8s side. This has to do with Kubernetes' maturity and people's confidence level, again most stuff are already automated, you could mostly use kubectl. But people are still learning, so this is why we are seeing that crossover. This also matches the nature of who is managing software on these clusters. It's a lot more operations people than it is developers."

—Kelsey Hightower, Principal Developer, GCP Advocate



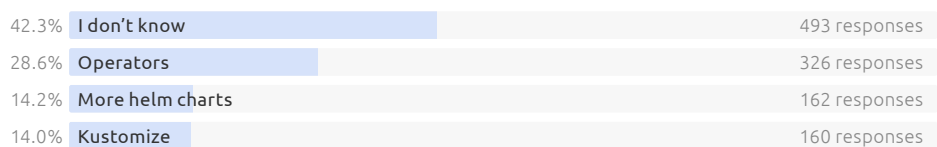
"I like Kustomize conceptually, but I do find it hard to use. And it looks like Helm is a lot more popular, which I think is because it's much easier to use. I would like to see Helm and Kustomize merge in some way, at some point, because it just seems silly to have two ways to do the same thing."

—James Strachan, CloudBees

24. What is the solution to too many Helm charts?

What is the solution to too many helm charts?

1141 out of 1166 people answered this question





“This is a problem in the ecosystem. It’s not that there are too many, rather that a great many Helm charts are of varying maturity. You cannot know what they do exactly without heavy due diligence. In the financial space, there needs to be a practice of vetting these from a security and maturity standpoint. The same applies to operators.”

–Ken Sipe, *Cloud Native Computing Foundation, Edward Jones*



“GitOps might be a way to manage Helm charts at scale. Solutions such as Jenkins X, Argo and Flux provide a way of configuring which Helm charts have been installed in which clusters using just source control gates and files in Git. But the thing that’s missing in these solutions right now is a standard Git layout of how multiple Helm charts should be configured in source code.”

–James Strachan, *CloudBees*



“Helm made it easy to deal with the current problem, which is too many YAML files. People wanted something simpler, an abstraction or a wrapper, but Helm only solved the Day One problem. An operator is going to go a little bit deeper into the full lifecycle management, active management and deal with healing and so forth. So that’s why I think operators are probably going to keep growing over time. People don’t know what the solution to too many Helm charts is, of course, because everyone’s still learning and practising.”

–Kelsey Hightower, *Principal Developer, GCP Advocate*

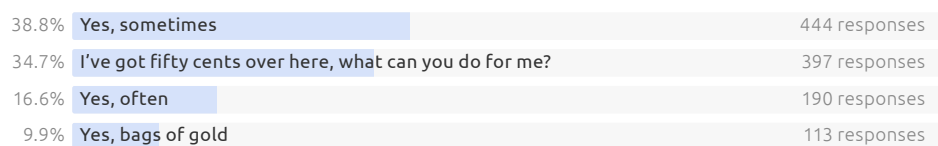
If you’re interested in learning more about operators and managing operations beyond Day One, here are some resources for you:

- [The Charmed Operator Lifecycle Manager](#)
- [The Charmed Operator SDK](#)
- [Step-by-step walkthrough of building a charmed operator from OperatorDay at KubeCon 2021](#)

25. Do you pay for support on your ops tools?

Do you pay for support on your ops tools?

1144 out of 1166 people answered this question





“This one is easy. If you want good tooling to live for a period of time, you’ve got to pony up.”

–Ken Sipe, Cloud Native Computing Foundation, Edward Jones



“I think people are willing to pay for things they don’t have time to build, use cloud provider services or buy a vendor’s provisioning stack. If they can build or download something for free, they’d probably be hesitant to pay for it, especially if it tends to work out of the box.”

–Kelsey Hightower, Principal Developer, GCP Advocate

26. Does your team use community-provided templates / playbooks /scripts / packages?

Does your team use community-provided templates/playbooks /scripts/packages?

1141 out of 1166 people answered this question

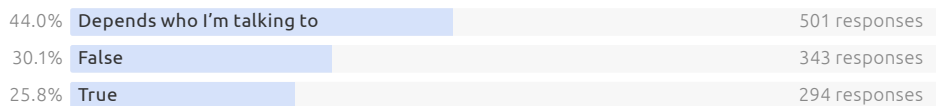


We see this as a strength of the open source community. If you have access to the source, [can run your own tests against it](#), and ensure security, then as a whole we’re able to solve the repetitive configuration tasks that are nearly the same, yet repeated again and again from project to project. [Collaboration over duplication](#).

27. Every line of every chart and YAML file

I believe that I should know every line of every helm chart and yaml file

1138 out of 1166 people answered this question



Like we stated above — there is power in working together to solve problems. That said — Trust is very very important. James and Kelsey weigh in:



“I think that for all the cloud infrastructure and cloud services, we do need to delve deep only when that is really necessary, but overall stay fairly high level and get things to just work. Otherwise, we will spend our lives in university learning every property of every Kubernetes resource and the new acronyms for all the different services, which come out by the thousands every year. Life shouldn’t be that complicated, therefore getting things to work is more important.”

–James Strachan, CloudBees



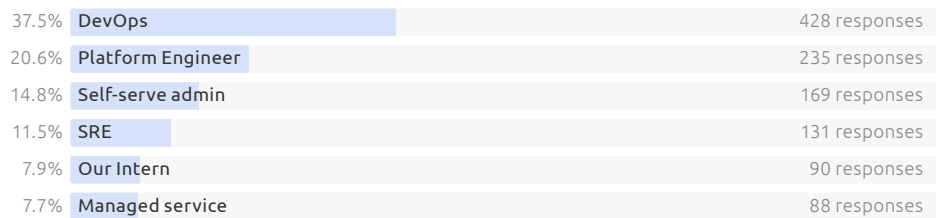
“This is where I would say someone in every organisation should know because Helm today can do pretty much anything in your cluster. So do you really not care about what’s in that Helm chart? I think this is really a matter of provenance. Did you get this Helm chart from a trusted actor, like your operations team or is it supported by an official vendor? Ok, that’s fine. If you’re just getting this from the internet and GitHub and you don’t know who wrote it, I think you should probably spend a little time checking it out in detail.”

–Kelsey Hightower, Principal Developer, GCP Advocate

28. In a typical organization, who manages the cloud infrastructure?

Who manages your organisation’s cloud infrastructure?

1141 out of 1166 people answered this question



“It’s really weird that “managed service” is so low given the high number of cloud-hosted Kubernetes usage. To me, this reads as if DevOps teams are expected to manage the entire cloud. Is that how businesses think of it? It’s not clear. I don’t believe 37.5% of the DevOps folks are all managing their own on-premise infrastructure. I think some of them are using tools like EKS. But maybe they are answering as if they are managing them because they own the relationship with the clouds.”

–Alexis Richardson, Founder and CEO, Weaveworks



“In the organisation I’m working with, they have a policy: you build it, you run it. So in some cases, you’d expect the app teams to be managing it — but they don’t really manage it. They manage a section or a subscription, or an organisational unit. When I think of somebody who manages cloud infrastructure, I think about somebody who spans multiple OU’s or has the right to create an OU. It’s really surprising not to see SREs more towards the top. Maybe it’s a lack of industry-standard on terminology?”

–Ken Sipe, Cloud Native Computing Foundation, Edward Jones



“When you take something as complicated as a running Kubernetes cluster and offload its management to individual teams — that’s just another side job to them — you’re going to have poorly configured, inefficient and insecure clusters that are governed differently from each other and are on different versions. I know some big organisations that started with this model and quickly realised it doesn’t work. So what they ended up doing is implementing a central governance model for those clusters, splitting responsibility between people who run and people who use the clusters. These are not necessarily the same team, mind you.”

–Tim Hockin, Principal Software Engineer, GCP

29. Top considerations for DevOps in cloud native technologies

What are the TWO most important questions that ops people should care about?

1141 out of 1166 people answered this question (with multiple choice)

46.2%	How secure is that thing?	527 responses
26.4%	How can we optimize resource utilization?	301 responses
24.9%	How can we optimize applications to reduce work/errors in Day 2 operations?	284 responses
20.5%	What resources should be allocated to the scenario?	234 responses
18.1%	Where should the application run?	207 responses
17.5%	Which software components should be included in the workflow?	200 responses
17.0%	What is happening in that YAML file?	194 responses
13.8%	What’s the fastest way to deploy X?	158 responses
8.4%	What is happening in that helm chart?	96 responses
7.1%	How should that scenario be integrated into the wider estate?	81 responses



“When people worry about the basics, i.e. security, costs, resources, observability and configuration, this is an indication that they haven’t yet crossed the chasm of full enterprise adoption.”

–Alexis Richardson, Founder and CEO, Weaveworks



“Security as the #1 concern — this might be the finance and government audiences coming out, combined with good DevOps practices. It might also be tied to current events — SolarWinds, the Colonial Pipeline hack. All of this might be bringing security higher in our minds.”

–Ken Sipe, Cloud Native Computing Foundation, Edward Jones



“If you look at the top four to five answers those are all about what the application should be doing. Security is always a thing. Then, now that we have all that visibility, from the existing tooling, the teams should be informed about what resources they use and we should try to get a much better cluster utilisation. The bottom part of responses I could label as “how to describe a deployment”. Should we use the .yaml directly? Should we use something like Helm? Do we know what’s going on in that Helm chart? Should we go get an operator?”

–Kelsey Hightower, Principal Developer, GCP Advocate

From the point of view of your humble authors, we believe that ops people should focus on these four important business decisions:

- Where should the application run?
- Which software components should be included in the scenario?
- What resources should be allocated to the scenario?
- How should that scenario be integrated into the wider estate?

To execute on these scenarios, we believe that software should be running your application.

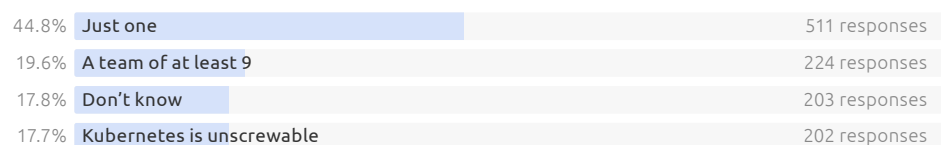
The Charmed Operator Framework (also called Juju) brings business thinking to the world of application management, making it much easier to have conversations between teams about the estate, and much easier to evolve capacity allocation over time: you can easily manage multiple compute environments and lower your costs of ownership. The separation of technical and commercial concerns means the operator is reusable in very different business settings. And reuse enhances the community value of the operator and increases software quality.

Most importantly, the Charmed Operator Framework is not a “language for configuration file management”. It is not a successor to Puppet, Chef, Ansible, Terraform or Salt. Juju removes the requirement to invest in deep institutional knowledge of configuration details altogether, by encapsulating community knowledge in operators that handle those details automatically, given a context. Moving beyond configuration management puts the focus on application management and resource allocation. It is the stepping stone to Model Driven Operations.

30. How many ops people does it take to screw in a Kubernetes?

How many ops people does it take to screw in a Kubernetes?

1140 out of 1166 people answered this question



While we were trying to make a reference to the classic joke of screwing in a lightbulb, respondents seemed to get our intention, but then Tim went on and dropped the mic:



“It takes a team of 9 to screw it in but just 1 to screw it up.”

–Tim Hockin, Principal Software Engineer, GCP

CNCF Landscape

31. Cloud native technology mastery

Which cloud-native solution space(s) have you mastered? Select all that apply.

1144 out of 1166 people answered this question (with multiple choice)

32.3%	Continuous Integration & Delivery	370 responses
24.4%	None — I’m pretty good at some of them, but master of none	279 responses
23.2%	None — I’m still learning	265 responses
22.8%	Application Definition & Image Build	261 responses
22.6%	Database	259 responses
22.6%	Automations & Configuration	258 responses
19.3%	API Gateway	221 responses
19.2%	Container Registry	220 responses
16.1%	Container Runtime	184 responses
15.4%	Ingress/Service Proxy	176 responses
14.2%	Observability	163 responses
11.6%	Cloud Native Storage	133 responses
11.5%	Security & Compliance	132 responses
10.8%	Scheduling & Orchestration	123 responses
10.7%	Serverless/Function-as-a-Service	122 responses
10.1%	Cloud Native Network	116 responses
10.1%	Streaming & Messaging	116 responses
10.1%	Service Mesh	115 responses
7.9%	Chaos Engineering	90 responses
7.9%	Key Management	90 responses
5.5%	Remote Procedure Call	63 responses
4.9%	Coordination & Service Discovery	56 responses

The [CNCF landscape](#) has grown to become the single reference point for cloud native technologies. The CNCF annual survey takes an in-depth pulse around these technologies and their usage across different categories. Our intention was not to repeat the work of the CNCF here, rather to ask respondents to share their level of experience and comfort working with the technologies from all the categories, as defined by the CNCF.

We are proud to say the cloud native community provided truthful answers, something that can be deduced from the fact that, besides CI/CD which got the majority of answers with 32%, most reported to be still learning at either a beginner or expert level.



“I think the really interesting part of this question is, literally everything in that list may be used on any project. And that’s what’s challenging and scary about the cloud. Depending on your use case, you might need one or more of everything on that list. And this ties back to training and lack of skilled people. Once you’ve made a few things in the cloud, you can quickly really build up confidence. But I can imagine a lot of people have never touched the cloud before and looking at that list they don’t know where to start. The hardest thing with the cloud is what technology and tools and services do you pick and in what order for a given solution?”

–James Strachan, *CloudBees*



“I think that the interesting thing that starts to match the rest of the analysis is that people, almost 50%, saying we’re still learning, which is similar to previous questions that showed concern around lack of skills and training. But, as I said before, they don’t realise they already have a lot of the required skills. Kubernetes is not where these skills originated. Observability isn’t new, it didn’t just start when cloud native came out. What is cloud native storage? All of the storage options that are being used right now originated 10 years ago, including S3. Low-level things like container runtime and container registry — we have Docker to thank for these — people feel more confident about because Docker has been around for a long time. Continuous delivery and integration. Those are things that existed before cloud native. So I think that’s why you see a higher response and confidence because people have been doing that since 2005. Overall, I would say that when we go down this list and see the new terms used for older concepts, many people lack confidence, because they believe that the fundamentals have changed somehow.”

–Kelsey Hightower, *Principal Developer, GCP Advocate*

Can we take a look at this from the exact opposite angle? Let’s not focus on the fact that not everyone is a master of particular technologies, but on the fact that there are already masters out there. There are 90 people here who consider themselves masters at chaos engineering. 163 at observability. 176 at ingres / service proxies. How many of them would it take to create charmed operators that cover the use cases of tens of thousands of Devs, DevOps, and SRE’s?

Clearly, based on the question about people using community-provided playbooks/charts/recipes and the interest in operators (described in the upcoming section), there is room in the cloud native space for experts to help the entire community to make significant advances by applying their knowledge.

Kubernetes operators

32. Would you trust an operator built by an expert?

I would be willing to use a MySQL operator developed by someone who has spent the last 5 years maintaining MySQL in production.

1134 out of 1166 people answered this question



“This is a really exciting area, operations at scale. Today, we are fundamentally seeing Helm, Juju, operators and other extension points. An enterprise app store for operators could be a thing in the future, but not just yet. With an enterprise app store in place, application developers could just choose the operator they want, pay as they go, and be much more efficient and productive.”

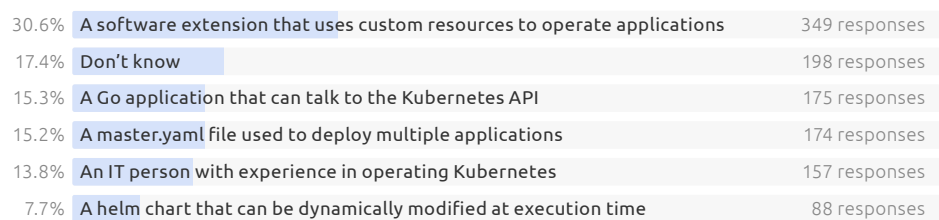
—Alexis Richardson, Founder and CEO, Weaveworks

Sounds interesting. Whether paid or free, access to the knowledge of experts, coded into operators, enabling application management at scale also sounds quite intriguing. We asked a few more questions about this.

33. Definition of an operator

Which of the following sentences most accurately describes a Kubernetes operator?

1141 out of 1166 people answered this question



Remember the question where we asked people for the definition of a hybrid cloud, and almost 80% of the respondents got it right? We're not there yet for operators — most likely because the domain is still being expanded on and clarified.

We like to define an operator as software that encapsulates a single application and all the code and know-how it takes to operate it on Kubernetes or machines.



“People need to understand Kubernetes operators if they want to do data in Kubernetes. These results show that we’re still early on the journey. Familiarity with K8s operators is a sign of maturity; it indicates how many people have started to think beyond just setting up clusters to leveraging the Kubernetes extensibility model. There are business decisions to be made around the extensibility model, such as how to work with it — in-house or outsource — and of course, limitations will stem from these decisions. I believe this whole topic will shake out as platform teams are going to be sophisticated users of operators and then app dev teams will consider them basically as add-ons.”

—Alexis Richardson, Founder and CEO, Weaveworks



“The first answer is the starting point for what people call “operators” in the space. I’m not sure that I would have answered this way. It’s really about domain-centric knowledge, which doesn’t require a CRD but it could use one. Ha — anyone who would claim ‘Go’ as a requirement is kind of jaded. There’s enough humour in the survey that maybe they were just being funny too? I’m not sure. There is support for tons of languages. Frankly, it’s just a controller, it could be Bash. There’s an opportunity to improve clarity here.”

—Ken Sipe, Cloud Native Computing Foundation, Edward Jones



“I agree, operators help by hiding complexity from the user and just run stuff on their behalf, so they do make Kubernetes easier. But there’s still quite a lot to know to get to that, unfortunately.”

—James Strachan, CloudBees



“I think of an operator as a control plane that you install into Kubernetes to give you an extended set of capabilities. It then boils down to setting the bar of what defines a really good operator. As a community, we need to raise the bar to say an operator has to deal with day two and day three concerns. It’s not enough to just go from Helm to the CRD.”

—Kelsey Hightower, Principal Developer, GCP Advocate

34. Experience with operators

What is your experience with Kubernetes operators?

1136 out of 1166 people answered this question

29.8%	Trying them out is on my to-do list	338 responses
16.8%	Getting comfortable with them	191 responses
13.7%	Using them for apps in production	156 responses
11.4%	Other	129 responses
7.4%	Most useful invention since pizza	84 responses
6.1%	Tried one, failed miserably	69 responses
5.5%	Don't care for them	62 responses
4.8%	Don't need them	55 responses
4.6%	We have security concerns	52 responses

What is your experience with Charmed Operators?

1135 out of 1166 people answered this question

31.0%	Trying them out is on my to-do list	352 responses
14.5%	Other	165 responses
13.6%	How are these different from the others?	154 responses
8.1%	Don't care for them	92 responses
7.3%	Getting comfortable with them	83 responses
6.2%	Don't need them	70 responses
5.9%	Using them for apps in production	67 responses
4.8%	We have security concerns	54 responses
4.5%	Tried one, failed miserably	51 responses
4.1%	Most useful invention since pizza	47 responses

The biggest takeaway from these two questions is that trying out operators are on to-do lists across the board.



"I think this just speaks to the maturity of these things. It's going to take time for these things to do more than just the initial deployment. Some people look at operators as a much easier interface to deploy complex software. But there are matters of trust when it comes to handling lifecycle like upgrades in production. Some don't even trust the people on the team that are supposed to be doing the upgrades. So I think it's going to be a while before we delegate that to the software."

—Kelsey Hightower, Principal Developer, GCP Advocate

Kubernetes advanced usage

35. What can't you do with Kubernetes

What things do you wish to be able to do in Kubernetes that you currently can't?

1136 out of 1166 people answered this question

22.5%	Make coffee	256 responses
18.2%	Automate the integration of cloud services with my on-prem applications	207 responses
17.3%	Have visibility around failure predictions of my Kubernetes clusters	197 responses
16.7%	Integrate legacy/MVC applications running on VMs with my containers	190 responses
11.9%	I have a monolith the company worships	135 responses
10.8%	I can do everything I need using Kubernetes	123 responses
2.5%	Other	28 responses

Yes, Kubernetes is the go-to platform on top of which modern applications are built. But can you do everything you want with it? Even things beyond your wildest dreams? Even coffee? The people have spoken, the next release should include pod support for at least a flavourful espresso. *Note: Canonical has proposed a session at Kubecon NA 2021: Los Angeles where we hope to show how we used Kubernetes to brew a fine cup (if it's accepted).*



"I can do everything I need using Kubernetes.' That's what we call Kubris."

—Alexis Richardson, Founder and CEO, Weaveworks



"I don't know why "make coffee" didn't get way more responses."

—Michael Hausenblas, Solution Engineering Lead, AWS

36. Selection criteria for container images

Which of the following do you value the most when selecting a base image of a container image? Choose 3.

1142 out of 1166 people answered this question (with multiple choice)

56.0%	Security — passed vulnerability and malware scanning	639 responses
38.8%	Stability — long term supported versions	443 responses
38.3%	Size — lightweight images	437 responses
37.8%	Compliance — with your company policies or a standard	432 responses
37.3%	Provenance — getting the image from a trusted publisher	426 responses
27.1%	Developer experience*	309 responses
22.9%	Ready-to-use — default packages and tools included	262 responses
21.7%	Base layer — preference for an Alpine, UBI, Ubuntu, etc	248 responses
20.1%	Price — lowest cost for pulling or running the image	230 responses

* Developer experience — frictionless usability of the image, Size — lightweight images, Stability — long term supported versions

Hardened, long-lived, trusted, and lightweight. An ideal base image is the building block for production-grade, secure, performant cloud native apps. As best practices evolve, the focus is now on curating what needs to be included inside OCI layers, at build time vs. at run time. According to our experts, while workarounds exist, there is currently no optimal way to address this challenge:



“When building our container images, we are porting old habits from VMs to the container world. We just take the OS and put it in the base image. I think there is a better way, over time, to start making smaller, lightweight images. Any one container we build the old way, is probably not using the largest part of the stuff it carries. We’re just carrying baggage, more bits to maintain and update. There is currently no efficient build toolset allowing us to benefit from the lightweightsness of a container image versus a full virtual machine. People mostly do what they already know and understand, but the resulting artifacts are far from ideal.”

—Kelsey Hightower, Principal Developer, GCP Advocate



“This is a ripe space for new exciting tech to solve an important problem; how to just take the stuff I need from a base image and throw everything else away. Take apt-get for example. I surely don’t need this at runtime, I only need it at build time, when I assemble my container. So why would it run as part of my base image? There is currently no good, secure way to do this. A real minimal image. There are solutions out there like BusyBox or Alpine — that actually includes BusyBox — but these are not enterprise-friendly. And then there’s Debian, Ubuntu, Red Hat. These are enterprise-grade distros, but still quite big as base images. Why do I need 75 megabytes of Python and other utilities when I just want to run my 12 megabyte Go program?”

—Tim Hockin, Principal Software Engineer, GCP

37. Stateful applications in Kubernetes

Do you run stateful applications in containers?

1141 out of 1166 people answered this question



Originally, containers were built to be stateless, flexible and portable. However, we see that only 21% use them solely for stateless applications, which also aligns with the latest data from the CNCF. The number that use stateful apps in production has dropped from [55% of the CNCF Report 2020](#) to 37% which is probably skewed due to the more varied sample we had in this survey — 50% of Ops team representatives that are using stateful apps in production were balanced by 63-68% of Devs, Managers and Executives that don’t. Also, the number of people evaluating or planning to use stateful containers in the next 12 months has grown from last year, which is another sign of the platform’s maturity. Even so, our experts bring a fresh perspective to these numbers:



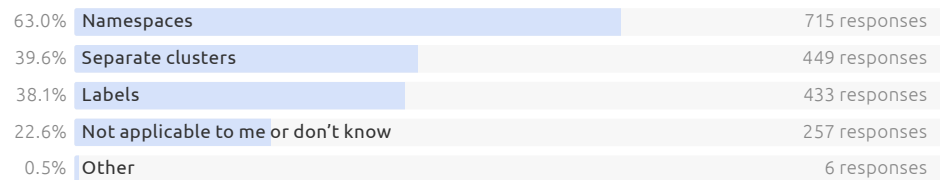
“This depends a bit on how people define stateful applications. All apps in the world have state of some sort. It’s more about where the state lives. Is it on a database? Is it on a local cache? Or do they mean state in a more traditional sense, something like Redis? Now, if Redis went down, are you out of business? The answer is probably ‘no’, you probably have a system of record somewhere else. And then you can just repopulate that cache and keep moving. I’d be surprised if 37% of those people are really dealing with the system of record in containers. Kubernetes is getting better, in that sense sure, but I need more questions to know if this is accurate.”

–Kelsey Hightower, Principal Developer, GCP Advocate

38. Kubernetes security: how to isolate your applications

How do you separate applications in Kubernetes? Select all that apply.

1135 out of 1166 people answered this question



Another sign of Kubernetes maturing is users asking how to address container security in a way that is viable for production in their organisation. Namespaces take the lion’s share when it comes to separating applications, while using separate clusters and labels are a close second and third.



“Interesting that people still think that namespaces are a matter of isolation. I think the challenge here is logical versus proper isolation in terms of tenancy.”

–Michael Hausenblas, Solution Engineering Lead, AWS



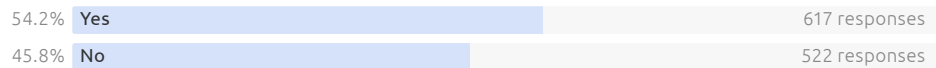
“This one is probably the biggest challenge people have. In the cloud, there are different ways to separate apps. On AWS, you can have an entirely separate account. In Google Cloud, you can have separate projects. What most people end up doing is trying to mirror the organisation in some way. Namespaces are a great way to have lightweight environments and according to the data that’s the default way. Using separate clusters, that’s very much the enterprise approach. I think the thing that’s missing here is separate nodes in the same cluster. There, you can create node pools for different apps. Like the database lands on these three nodes and the main app is on a different set of nodes. That’s kind of an intermediary step before going all the way to separate clusters. There’s no one way to guarantee separation of your apps, this depends a lot on your workflows.”

–Kelsey Hightower, Principal Developer, GCP Advocate

39. Kubernetes like a Pro. Highly-available. Air-gapped. Offline

Are you running High Availability Kubernetes clusters?

1139 out of 1166 people answered this question



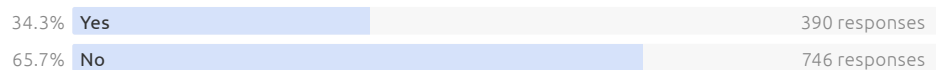
Are you running Kubernetes in an air-gapped environment?



1138 out of 1166 people answered this question

Are you using Kubernetes in an offline environment?

1136 out of 1166 people answered this question



As people start to get more familiar with Kubernetes, there are trends towards a more advanced use of the platform; users deploying Kubernetes in a highly available architecture to ensure full uptime for production and even extreme cases where they look to be independent from an internet connection while running their clusters. Here's what Kelsey Hightower had to say about all this:



"I think the 54% is using high availability where that's easy to do, like a cloud provider's distribution or a managed Kubernetes offering, where you click a button, and have it spread across multiple zones. If it's not easy, then I doubt people will put in the extra effort needed to run an HA Kubernetes — it can be a huge undertaking for people who've never done it before.

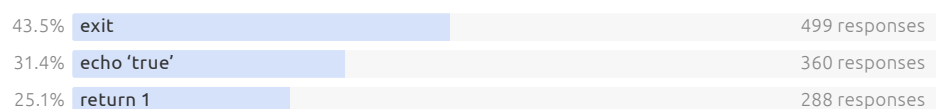
For those unfamiliar with air gap it's like saying "I want to never depend on the internet to run my Kubernetes cluster". I think that's really reserved for extreme cases like running in a department store, a retail environment or in a government setting. The numbers here loosely reflect reality, my guess is more like 85-90% of people are not running in an air-gapped or an offline environment."

—Kelsey Hightower, Principal Developer, GCP Advocate

40. Enough about Kubernetes. Leave me with my bash scripts.

Bash is the only programming language anyone needs.

1147 out of 1166 people answered this question



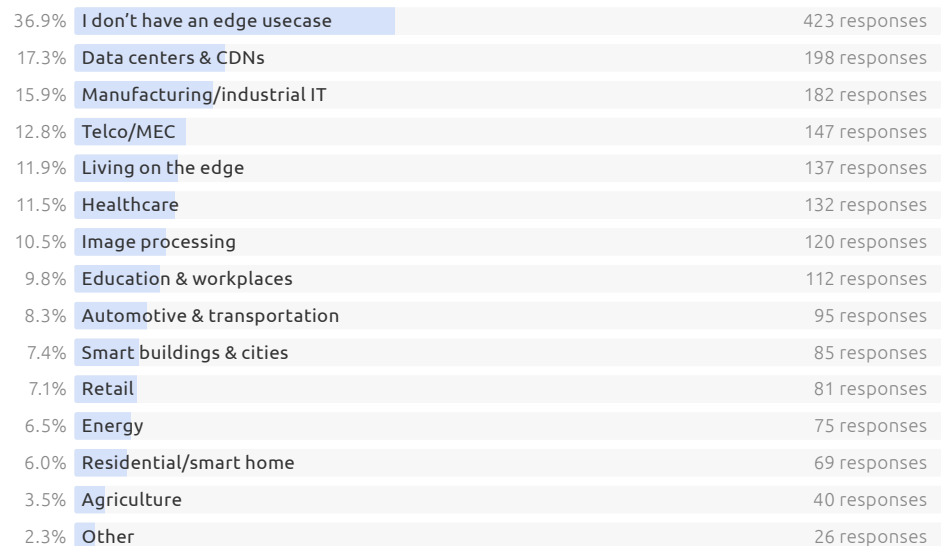
We respect everyone's opinions and preferences :)

Edge computing

41. Kubernetes at the edge

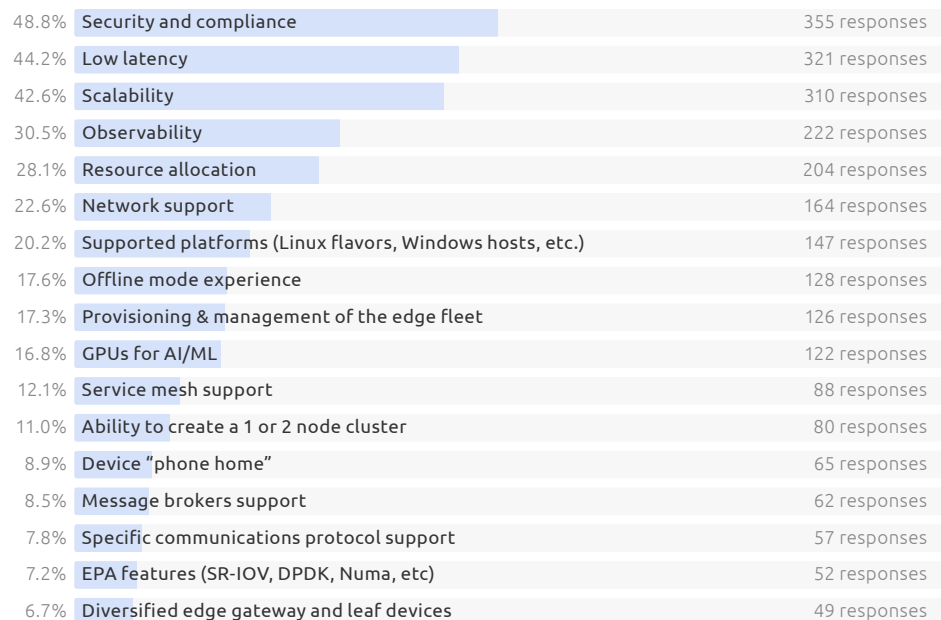
What is your edge use case? Select all that apply.

1147 out of 1166 people answered this question (with multiple choice)



What are your requirements when it comes to implementing an edge strategy? Select all that apply.

727 out of 1166 people answered this question (with multiple choice)



Wrapping up the report with a very hot topic that is mostly looking towards the future of cloud computing: bringing applications closer to the consumer, where scale, latency and operations across geographies are the main challenges. Various industries have already made that investment and their representatives shared the main requirements for Kubernetes around the space. More than anywhere else, security and compliance is the top priority at the edge.



“Minimal Kubernetes distributions, like MicroK8s and k3s, have revolutionised Kubernetes for the edge. Think about the telco or healthcare industries; they are currently running embedded software in a way that is very inefficient. Now they can transform their business using microservices on Kubernetes at the edge, with lower footprint and optimised resource allocation without impacting performance.”

–Karthikeyan Shanmugam, *Digital Solution Architect, HCL*



“For brick-and-mortar locations that have a computer inside that runs business apps, the previous choices were a couple of Linux machines or virtualization. My guess is that the sample of people that already made those investments are looking at Kubernetes to be a lighter-weight implementation of sharing compute inside these brick-and-mortars. All the requirements listed here are typical things one would want in a data centre. I do think Kubernetes will bring a lot to the table that didn’t exist before, such as observability or security and compliance at the app level and improved resource allocation. As a result, it will be easier for businesses to start provisioning and managing their fleet, keeping VMs as simple as possible, and to focus on the apps at scale with K8s.”

–Kelsey Hightower, *Principal Developer, GCP Advocate*

Conclusion

The Kubernetes and cloud native community is set on a trajectory to redefine the way software is built, deployed and operated in the long term. It is a paradigm shift from traditional approaches, so it makes sense that the majority of users and businesses are still considering themselves as students rather than experts.

There is surely a Kubernetes test environment in every organisation that leverages software as the primary way to run its business, and it is very optimistic to see that people have started thinking about solving Day 2 to Day 1000 problems at scale and in a standardised way. The cloud native community is committed to improving users' experience while onboarding to Kubernetes and helping them address their challenges and business requirements by improving the technologies and sharing best practices.

The Canonical team wants to thank all community members that contributed their time, experience, and answers to the survey, which made building this report possible, and to the industry experts who enhanced it with their insightful commentary. We are always keen to get your feedback to improve, and we're happy to ask questions to the community on your behalf. So, if you like to, you can [send us your questions and feedback](#).

If you feel that your industry was underrepresented or if you want to contribute to the community by sharing your insights, remember the survey is still open.

[Take the survey](#)

This report will be published bi-annually, a few weeks after every KubeCon event. The next survey iteration is planned for the week of KubeCon NA 2021, and the new report will be launched in November 2021.