

Embracing AWS CDK (Cloud Development Kit) for Next Level Devops

*No more elaborate console clicking sequences or rigid and repetitive declarative scripts - now we just write code to deploy Apps, Server, Network and Security - **infrastructure as code** in the truest sense.*

In the tech industry, superlatives about ground-breaking innovations or applications are commonplace, probably annoyingly so. For this reason, it's often best to be cautious about proclaiming the new next best thing, and our team at 2pi Software have muted our exuberance about our deep embrace of AWS CDK for almost a year now...but today that changes.

For those of you who may be unfamiliar with CDK, AWS describes it as 'The AWS

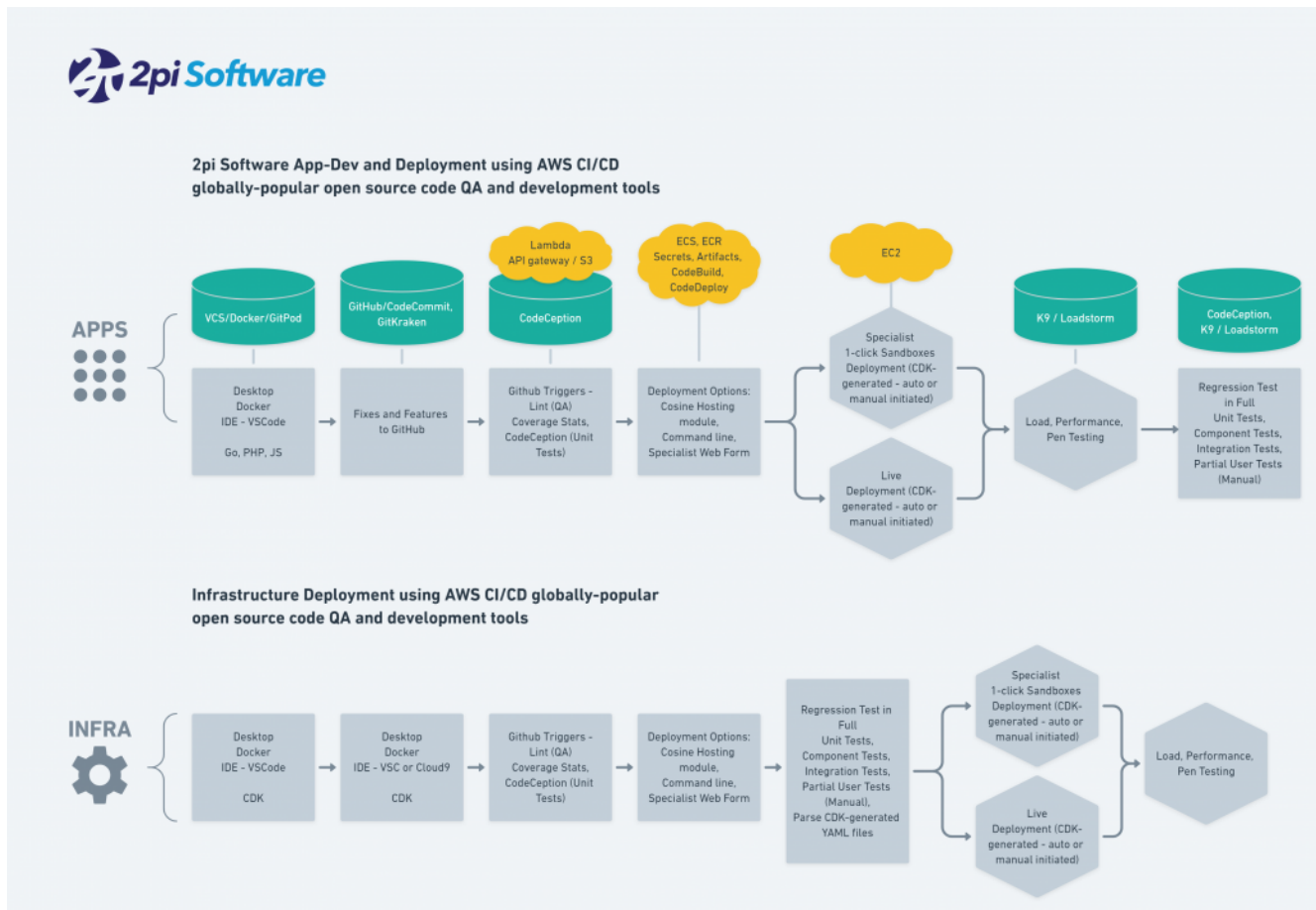
Cloud Development Kit (AWS CDK) is an open source software development framework to define your cloud application resources using familiar programming languages.'

As a technology business 2pi Software has been cloud-focussed from day one and were early adopters and proponents of the Amazon cloud suite, AWS. Over the years, the team has built and managed a range of complex systems for many corporate and government organisations using a whole host of helpful technologies - Cloud Formation Templates, Ansible, Terraform, Boto, Bash, Powershell and Python amongst others. In early 2020 our cloud practitioners' attention was caught by emerging dialogue around AWS CDK and the ability it offered to create and manage cloud infrastructure in an almost identical way to that used to create apps, and even allowing the same tools (e.g. the VSC IDE most notably) to be used.

Further investigation followed, and thanks to some excellent support and instruction from the Canberra AWS team, especially Brian Farnhill, by mid-2020 we started to experiment with AWS CDK in earnest and built real world solutions to our infrastructural requirements. Normally when a new technology or approach to technology is introduced, human factors and tech preferences often lead to hesitation or inertia from highly-skilled devs and ops people before they will jump into the unknown, as it were. But, fascinatingly, with AWS CDK, we experienced a quite rapid embrace of it, and indeed, once our staff saw the speed, convenience and power of AWS CDK, and its integratability to our CI/CD pipeline, their enthusiasm warmed quickly.

To cut a long story short, the below diagram, reflecting the unified enthusiasm of our dev and ops teams for AWS CDK, best summarises the significant improvement in all aspects of our app and infrastructure deployment that use of CDK facilitates. **In essence, and for the first time, the steps involved in app creation,** (ie writing and refining lines of code in popular programming languages such as Typescript or Python) **are near-identical to the steps involved in infrastructure creation,** (ie also writing and refining lines of code in popular programming languages such as Typescript or Python). In our humble

view, this is nothing short of spectacular from an efficiency, speed to deployment, and organisational management perspective.



Zoning in on why it works so well for 2pi Software to use AWS CDK, essentially, the devs never have to leave their trusty IDE and command line interface environments to do stuff - CDK frees them significantly from having to dive into the AWS Console and *click themselves into oblivion* to create or modify infrastructure to meet their needs, it's just another push or pull to/from Github, basically.

And the ops people, although there is a learning curve in picking up skills in the basic toolsets commonly used by devs, they very quickly see the advantages of working from the IDE and fashioning short concise AWS Typescript files instead of, respectfully, page after page of declarative Cloud Formation Template script. And they don't have to be code geniuses to navigate their way around Typescript to make meaningful changes - for the most part, simple code flow with branching

and looping does the job, there is generally no requirement for ops to compose complex object oriented structures or clever design pattern implementation.

This overlap between how coders and sys admins can work has another enormous benefit in that they start to talk the same language and bring shared efficiencies to the whole process of marrying app deployments to infrastructure in a fit for purpose optimal manner. Even the occasional disagreements are better, or at least better communicated!

As a general aside, discussing separateness between devs and ops possibly runs a little counter to the devops zeitgeist of recent years. While it's clear that the devops mantra is relevant and practical and directionally sound, it is, based on the 2pi Software experience, not quite as clear cut as many would have it. The realities of the industry are such that there appears to be a continued perpetuation of dev-inclined and ops-inclined humans in the sector and although the overlap between both flavours is increasing, the fully merged state is not quite on the horizon yet. That said, AWS CDK, and any trend towards greater true infrastructure as code brings next level devops closer and closer.

With CDK the icing on the cake for 2pi Software is that both app code and infrastructure code merrily progresses along the various waypoints of the CI-CD pipeline from desktop environment to git repo to docker image to regression testing, staging and production. In terms of modern software development and deployment to the cloud, the addition of AWS CDK to proceedings is entirely game-changing.

For 2pi Software, the true impact of this in terms of better serving our customers is the ability to fully automate deployment of our flagship products, the open source Cmfive (cmfive.com) business framework, and Cosine CRM. All steps along the way, from metal to executable, coded in Typescript and Lambda-triggered Python, enabled by a handful of commands representing an enormous step forward for the business and its cloud capabilities.

If you would like to see more detail on this process and a basic explanation of our approach of maximising the power of CDK, please check out the YouTube video from our April 2021 'CDK-ing the Monolith' presentation at the AWS User Group Meetup in Canberra below.