A Story of Future's Workspaces: Cloud Workspaces

Anastasia Protopapa, Parumita Saha

April 2019

The last years the culture of software development has changed dramatically. Today, people and organizations expect fast and reliable delivery of the software they use. Organizations wanting to survive the competition and deliver reiable software in a fast and secure way, have turned on new ways of working: Agile Software Development and DevOps. Additionally to those transformations, many companies deliver their software through the Cloud, which removes the burden of managing computer system resources and focusing solely to the application and the business logic of the problem each organization tries to solve. We can say that all the software industry has reevaluate their ways of working and infrastructure and we believe that the next step to reevaluate is their development workspaces.

Employees and specially software developers expect mobility and BYOD (Bring Your Own Device). Workers want access to business applications and data on-the-go from any device, including their own personal devices. At the same time, organizations are increasingly focused on reducing IT budgets, protecting privacy and intellectual property, and complying with regulatory requirements. Workspaces as a service is the answer to all the previous demand. Moreover, is a complementary tool that will help companies and large organizations on their main purpose to be more productive.

On the following sections we present some of the many tools that exist out there: Cloud9, Codenvy, Codeanywhere & CodeTasty. For you to get a small idea on the number of available tools and their reputation among users, you can jump to the 32 Best Cloud IDEs on Slant.

Table 1: Cloud9 Pros & Cons

Cloud9	
Strengths	Weaknesses
Editor highlighting and code autocompletion	Credit card on sign up
Built-in terminal & Debugger	Expensive
Debug with breakpoint for Node.js	No build-in Java builder & runner
Support for most DBs	No ‘chroot‘ functionaliy
Connects with BitBucket, GitHub, GitLab	No subdomain options
Hosting on premises	
Support for popular languages	
Shareable Configured Workspaces	

Cloud9

Cloud9 is one of the first browser-based IDE was founded in 2010 and acquired by Amazon on 2016. It is written in JavaScript, and uses Node.js on the back-end while the editor component uses Ace.

The IDE comes with features such as syntax highlighting, code autocompletion, live preview, built-in terminal, and a debugger. It can connect to existing code repositories hosted on Github, BitBucket, Gitlab and others. It can also be configured to talk to a database server. A configured workspace can be easily shared with other team members. The first step that differentiates it from a Desktop IDE is that developers can connect Cloud9 IDE to their own VM via SSH, which provides immediate access to the dependencies and runtimes. Moreover, applications developed in Cloud9 IDE can be deployed to a variety of targets including Heroku, Microsoft Azure, Google App Engine and Cloud Foundry. Table 1 summarizes the good and the bad of Cloud9.

It’s is generally accepted that Amazon through AWS is evolving fast to become the largest platform company of our times. Many argue and support that EC2 is the new OS, S3 is the internet file system, Aurora is the new RDBMS, and finally AWS Lambda is the polyglot runtime of the internet. It is obvious that Amazon may not be owning an operating system or a language, but is transforming to be the meta-OS and meta-platform company. Amazon’s acquisition of Cloud9 IDE will result in many interesting scenarios. AWS lacks the right set of tools for writing IAM (identity and access management) policies, CloudFormation templates, AWS IoT Rules, Chef Cookbooks and Recipes for OpsWorks, Stored Procedures for Aurora, and an integrated shell to interactively run CLI commands. Cloud9 would fill major gaps that exist in current AWS offerings. With all that on mind, we can say that Amazon will eventually move every aspect of coding and scripting to Cloud9. Developers will be able to write, debug, test, and publish AWS Lambda functions from a single, unified environment. With the integration of AWS CodeCommit, Cloud9 would automatically commit the source code. AWS CodePipeline will bring CI/CD and build automation capabilities to the platform while AWS CodeDeploy will push the artifacts to EC2, Lambda, or ECS.

Cloud9 will help Amazon establish its dominance in the cloud market. This move from Amazon has challenged Microsoft in multiple ways since Visual Studio doesn't have an online, browser-based version that tightly integrates with Azure. The acquisition is also an endorsement for other cloud-based IDEs such as Codenvy that we present in the following sections.

Codenvy

Codenvy is a cloud workspace for development teams. Developers can use it to code, build and test applications. It is powered by Eclipse Che. As it's visible on their site, on 2017 was acquired by RedHat in order to combine resources to create an agile development platform for OpenShift-powered applications. Codenvy has established strategic partnerships with Red Hat, Docker, Microsoft and SAP and uses its seat on the Eclipse Foundation board to drive the cloud development tools top-level project.

The below packages are offered:

- **Codenvy.com** - A hosted cloud IDE with support, SLAs, and hardware.
- **Codenvy Enterprise** - Enable organizations to code, build, test and deploy applications, on their own servers.
- **Codenvy ISV** - Drive and measure technology engagement of published SDKs and APIs with promoted Factories, monetizable plug-ins, and IDElets. A Factory is a way to launch a temporary code, build,test, debug workspace with policies. An IDElet is an embeddable code, build, test, debug workflow that can be inserted into another product.
- **Codenvy Platform** - A cloud IDE engine to provide developers a way to develop, test, and run tooling plug-ins and applications.

The Condevy SDK is similar in structure to the Eclipse Platform and is based on Eclipse Che, but engineered for a cloud environment. It also provides support for developing plug-ins for build, run, test, and debugging workflows, which typically operate outside of the IDE itself. The Condenvy IDE supports most of the major technologies like Java, Spring Framework, JavaScript , Ruby on Rails, Python etc. Additionally, the available platforms with which is can connect are: Amazon Webservices (AWS), Elastic Beanstalk, CloudBees, Google App Engine, Heroku, Red Hat's OpenShift .

Codenvy enables every developer to get cracking with the code without having to spend time setting up, installing, and configuring their development environment. The concept of workspace provides a sandbox that comes with everything that a developer needs. This dramatically reduces the time to modify

Table 2: Codenvy Pros & Cons

Codenvy	
Strengths	Weaknesses
Plug-ins for build, test, debug, run workflows	Lack of keyboard shortcuts
Conectivity with various cloud provides (e.g Heroku, AWS, etc)	No *ssh* or *ftp* mode
Effortless Java support	No collaborative editing
Portable Workspaces	No two-factor authentication
Beginner friendly	
Offline support	

existing applications or setting up a development environment for new applications.

One of the key differentiating factors of Codenvy compared to other solutions provided out there, is agile development. Organizations can connect to Jira, Github, Jenkins, and other tools that enable continuous development. Developers work on a feature branch in a shareable workspace which can be opened to other team members for immediate feedback and testing. Workspaces can be cloned for enabling instant reviews without disrupting the original workflow. Behind this, Codenvy uses Docker to make workspaces portable. Each workspace consists of a source code repository, project-specific artifacts, and multiple runtimes packaged as Docker containers. This tiered architecture makes Codenvy modular and portable. Through the concept of workspace recipes, developers can easily replicate environments within Codenvy's cluster or even on-premises. Even if the developers don't use Codenvy IDE, they can still take advantage of workspaces. Each Docker container comes with support for SSH and Git to provide direct access to the command line if needed. Last but not least, Codenvy creates the workspaces from the Docker and compose files of the application in production, which results to identical environment in development as well as in production. Table 2 summarizes all the characteristics of Codenvy.

Codeanywhere

Codeanywhere is TechStars startup on the mission to take coding into cloud. It's cloud based code editor and collaboration platform, pitched as "Google docs for developers", has all the features of a Desktop IDE but with additional features only a cloud application can give you. Another cool feature is "coding on the go" from all mobile platforms.

Codeanywhere Technologies focuses on microservices and is updating its API management portfolio to help developers create and deploy microservices, as well as manage the APIs used within microservice architectures. Some of the basic features provided are: expanded microservice capabilities for CA Live API Creator, Docker container deployment operations and an expansion to the API

Table 3: Codeanywhere Pros & Cons

Codeanywhere	
Strengths	Weaknesses
Integration with Dropbox, GitHub, FTP & SFTP	No debug
Connects with BitBucket, GitHub, GitLab etc	No good UX
Provides mobile application clients	Questionable 2-factor auth
Integration with DigitalOcean	

management portfolio to address management, security and advanced monitoring for APIs. Codeanywhere provides integration with services like Dropbox, GitHub, SandBox, FTP, SFTP which results on providing the developers with the freedom to access and manipulate their code on almost every server or repository. Table 3 summarizes the experience with Codeanywhere.

Codeanywhere provides a set of predefined development environments:

- **MEAN** is a fullstack JavaScript platform for modern web applications. MEAN is composed of and stands for; Mongodb, Express.js, Angular.js, Node.js. This framework allows for rapid set up and deployment of web apps, websites, web services and APIs.

- **Laravel** is one of the most popular open source PHP web application frameworks. Laravel is designed for the development of model–view–controller (MVC) web applications.

- **Sails.js** makes it easy to build custom, enterprise-grade Node.js apps. It is designed to mimic the MVC pattern of frameworks like Ruby on Rails, but with support for the requirements of modern apps: data-driven APIs with scalable, service-oriented architecture.

- **Wordpress** is a free and open-source blogging tool and a content management system based on PHP and MySQL.

- **Symfony** is a PHP web application framework for MVC applications. It is created as a set of reusable PHP components. The standard foundation on which the best PHP applications are built. Choose any of the 29 standalone components available for your own applications.

CodeTasty

CodeTasty is a fully featured cloud IDE in the cloud, and is quick and simple to set up and of course the last cloud workspace that we will review on this article. It enables you to code and collaborate no matter where you are and what device you are working on.

Table 4: CodeTasty Pros & Cons

CodeTasty	
Strengths	Weaknesses
User friendly & Young Code auto-completion etc	not tested product

It is fast and scalable. You can easily handle hundreds of thousands of files in your workspace and hundreds of thousands of lines of code in the editor. CodeTasty supports more than 40 languages and of course all the major languages: PHP, JavaScript, Python, Java, Ruby, C and of course Typescript. The editor itself is a full virtual development environment complete with terminal and output windows. CodeTasty is free and fully functional, though paid tiers are available for more project options and team collaboration.

Cloud Workspaces, IT & Modern Business

The first question that pops up when someone is firstly introduced on Cloud Workspaces is: "What is the difference between a Cloud Workspaces and a Desktop Workspaces ?"

Well, on each of the above tools that were presented, we can see that the primary technical difference between the two types of IDEs is that with a desktop IDE, the IDE vendor has an expectation that the packaging, build, test, and runtime environments are installed and managed outside the tool itself. This is not always the case, as some really advanced IDEs take care of installing these additional components onto the host machine, but it's not usually the case. With a cloud IDE environment, the IDE exists entirely in the cloud; however, usually the cloud IDE vendor must also provide the build and runtime environment in a hosted location in addition to the executing IDE. Moreover, the developers have access to those environments anytime and from any device.

But we should admit that this distinction offers both promise and challenges. The promise for cloud IDES is that being entirely in the cloud, the cloud IDE vendor can provision more components, do it faster, and potentially reduce failure rates from configuration. The challenges include the additional overhead of managing a workspace, which now includes an IDE in addition to projects, code, builders, and runners.

By taking into consideration that cloud workspaces provisioning is managed exclusively by the provider, an organization and it's employees benefit in the following ways compared to the traditional workspaces:

- Removes workload from the IT department since it's not their responsibility to maintain machines software that employees need.

- Employees such as developers can bring and use their own device. This has many benefits apart from the obvious, which is employee satisfaction, such as less costs in the organization. On the one hand, hardware costs, such as providing machines for employees is not needed anymore. On the other hand, Cloud workspaces reduce the costs on software licenses, since those also are managed by the provider and tha organizations pays only what it uses for as long as it uses it.

- Eliminate the threats to the organization's data security posed by fear of device theft and unwanted cyber attacks with remote access control and role delegation, since again this is handled in a centralized manner by the provider.
- Employees can work literally from anywhere with no overhead of VPN solutions.

But compared to on-premise IT solutions, there is one area that cloud-based solutions will carry higher risk: **Internet connectivity**. Cloud services require a highly reliable Internet connection with sufficient bandwidth to accommodate the the users at a location. Today, the quality of business-grade Internet services in most locations is excellent and affordable. Although Cloud Workspaces are engineered to be delivered over low-quality networks, bigger is better and a backup connection is a good idea where service is mission-critical. This has led many providers, like Codenvy, to shift directions on Hydrid solutions.

Craig Muzilla, senior vice president, Application Platforms Business, Red Hat, referring to the Cloud Workspaces says:

" Thanks to the increasing push towards digital transformation and the use of technology platforms, including apps, as a strategic business advantage, the role of the developer has never been more important. But, accelerated innovation through agile development requires news approaches and tools. "

But the main question remains: "How can Cloud Workspaces enable business productivity?"

With a majority of the workloads moving to the cloud, development environments and workspaces are also following the same path. The evolution of JavaScript-based browser frameworks delivering powerful user experience, combined with the power of containers are driving the next generation of IDEs. We will also argue that the demands on the development teams to release more frequently are increasing faster than their ability to manage the provided development infrastructure. The only way these development teams are going to become more productive is if the development infrastructure becomes transparent and is managed for them. As DevOps helped enabled powered organizations to deliver fast and reliable software after the commit, Cloud Workspaces will enable and power the developers to do the same, addressing the development infrastructure needed before this commit.

On the most basic level, Cloud workspaces can be used to reduce the scope of the desktop lifecycle management tasks. On a more advanced level, Cloud workspaces is the *Infrastructure-as-Code* of the IT by providing faster and effective procedures for supporting the overall IT infrastructure, including physical servers and virtual machines. In a real world examples, Cloud workspaces can be used to train employees, giving them access to a required set of tools for a limited time period. For employees working from home, it provides the required security by not storing data on end-user devices. Outside the IT community, Cloud workspaces can be used by schools & colleges to give students access to a WorkSpace for the duration of an academic year, and instances can be terminated at the year's end.

Closing this report, we would like to point out that the fact that all major organizations, which lead on the Cloud sector, are investing on Cloud Workspaces helps us to support the statement that they are the future of workstations and software development tools.

References

'Codeanywhere.com'. https://www.codeanywhere.com/, a. Online; accessed 21 May 2019.

'Codenvy.com'. https://codenvy.com/, b. Online; accessed 21 May 2019.

'Aws Cloud9'. https://aws.amazon.com/cloud9/. Online; accessed 21 May 2019.

Carlos Delgado. 'Nine Best Cloud Ides'. https://ourcodeworld.com/articles/read/276/top-9-best-web-free-based-integrated-development-environment-cloud-ide, 2016. Online; accessed 21 May 2019.

Helen Beal. 'Red Hat to Acquire Codenvy to Extend Devops Tools Capability'. https://www.infoq.com/news/2017/05/red-hat-acquires-codenvy, 2017. Online; accessed 21 May 2019.