WHITE PAPER

CollabNow Video Messaging on Kubernetes

By CollabNow – Delight Remote Collaboration

August 22

CollabNow

Table of Contents

EXECUTIVE SUMMARY

With real-time video streaming applications, no technology has been developed yet that can overrule these applications while implementing multi-element video streaming data-based pipeline. For harnessing the value from video streaming in real-time without conceding the employee location, developers are required to turn to several technologies – and using the Kubernetes cluster is one of them – relying on the need for data processing, and ingestion, analysis as well as serving. The truncation of conventional software release has its roots not only in technological developments like rapid growth of containers, and cloud platforms along with microservices-based structures but also in cultural developments having mobile-enabled and tech-savvy users that expect new features, quick to respond and fast fixing of bugs in developing products.

CollabNow is a video collaboration and conversational intelligence platform that fosters organizations to work smarter, faster, better, and more secure without compromising employee location. It possesses various features along with video messaging features. However, CollabNow uses the Kubernetes cluster for communicating these video messages from one user to others. This white paper is about Kubernetes and how CollabNow utilizes its video messaging feature on Kubernetes.

INTRODCUTION

CollabNow is a cloud-based video collaboration and insights platform designed to provide a scalable and reliable productive and engaging collaboration from any device. Our business is focused on improving the productivity and efficiency of remote and hybrid teams by providing a natural collaboration experience as close as possible to enterprise-level security, compliance, and manageability. Building trust and facilitating cost-effective seamless virtual collaboration to improve business outcomes is our goal.

CollabNow is a patent-pending SaaS-based video collaboration and insights platform hosted on Amazon and Oracle Cloud. CollabNow's distributed and microservices architecture with enterprise-level broader security provides a scalable and reliable experience for video, audio, and content to accelerate productivity, collaboration, and innovation.

Why CollabNow?

CollabNow has most interesting features that other market competitors does not posses.

Collaborative Features	*	🟨
Multi User Multi Screen Sharing, Zoom In/Out, Play and Pause Screens, Trim & Share Recording	X	✓
Video Messaging, Conversational Analytics	X	✓
Digital Agreement Rooms, 360 Degree View of Live Events	X	✓
Enhanced Security & Access Controls	X	✓

* *Zoom, Microsoft Teams, Google Meet, Cisco WebEx, Vimeo, Vidyard*

CLOUD-NATIVE

As cloud-based infrastructure, platform and tooling have developed as well as mature, and large-scale cloud deployments and migrations well-versed in the community of software development, the broader and extended approach known as Cloud Native has also emerged. At a significant level, Cloud Native apps are supposed to be containerized, carved up into various microservices, and are also designed to be deployed dynamically and proficiently executed by the orchestration processes such as Kubernetes.

For an Application to be Cloud Native

For an application to deploy, execute as well as manage the cloud Native, it should run numerous cloud native practices. For that purpose, Cloud Native-based application must:

- Be able to expose the health check endpoint in such a way that systems of container orchestration can analyze the application state and then react accordingly.
- Be capable of punishing the logging as well as telemetry data continuously, so that it can be stored and investigated by the systems such as Prometheus and Elastic search for metrics and logs, respectively.
- Have the ability to degrade in a kind way and cleanly tackle the failure so that orchestrators may recover by replacing or restarting it via fresh copy.
- No need for any human involvement to initiate and execute.

CollabNow on Kubernetes

For illustrating the significance of implementing the best practices of Cloud Native entailing containerization with the microservice architecture, we have been explaining a video messaging-based communication application known as CollabNow that provides on-prem and cloud-based video collaboration and insight services.

Throughout this paper, we have rationalized CollabNow through:

- Decomposing the business functions of the application into various microservices.
- Containerizing numerous elements into portable as well as deployable pieces.
- Utilizing the Kubernetes cluster for scaling and also deploying stateless microservices.

With our video message sharing monolith app CollabNow, we observe that initially WebUI, video management along with the user management business function are integrated into a single codebase where the segmented components summon one another through the function cells. The codebase is then constructed, assessed, and then deployed as a single unit, which is then scaled vertically or horizontally. Since CollabNow can combine more than one user at a time, thus codebase turns out to be large as well as complex. Even though code varies and bugs get fixed in minor, the whole app requires to be rebuilt or re-deployed, emphasizing a slower iteration cycle.

MICROSERVICES

MicroServices is known as the style of software architecture that supports various granular services where every service performs a single business function. Each of the microservice is said to be an independently deployed and self-contained piece of a complex application that relates to various other components, usually through REST APIs.

Microservices have emerged as a hybrid of various software development ideas and trends. A few of them are Service Oriented Architecture (SOA), containerization as well as DevOps which are considered to be more recent while others such as Unix Philosophy summon decade-old ideas and principles. Microservices were initially pioneered by companies like Amazon, Google, and Netflix, however, microservice architecture has turned out to be commonplace now in applications of all sizes.

CollabNow MicroServices Architecture

Considering CollabNow as microservice-oriented architecture, we have observed that single business functions like video communication management, as well as user administration, have been segmented into discrete microservices communicating through REST APIs. Here, each of the microservice is capable of using a data store for the appropriate data type that it would be handling. For instance, a video management microservice can utilize the cloud object store for videos with the rational database management system (RDMS) for the metadata. After that, the service can also be developed, analyzed, assessed, and deployed separately from various other services. By enabling high-level development agility, teams leveraging microservices can make use of the cloud infrastructure proficiently.

The microservices architecture is considered to be the crucial part of Cloud Native app development since these applications are comprised of containerized, and portable microservices which are not only disposable but also scalable. By segmenting the monolithic app into various self-contained, fine-grained services, where every single microservice can be scaled as well as distributed independently across the regions, and also implemented across several cloud platforms, Microservice allows to perform fast iteration and enhanced flexibility in executing the app on cloud infrastructure.

CONTAINERS

Up till now, we have made a discussion regarding the architecture and approaches underpinning the applications of Cloud Native. Now, we'll discuss the core technology that would enable the developers to deploy the broader architectural designs using containers.

What are containers?

These are known as the way of packaging the applications using the desired dependencies along with libraries in portable as well as easily implementable format. Once it is launched, the packages deliver the predictable but consistent runtime environment for containerized applications. Benefiting from the isolation feature of the Linux Kernel, the implementation of runtimes of containers delivers the sandboxes along with the resource-controlled execution environment for the apps.

Application Dockerization

Application containerization through Docker includes writing the manifest of the container image known as Dockerfile. The file explains the way to construct the container image via defining the initial source image and then outlining procedural steps necessary for installing and deploying the dependencies like the libraries and language runtime, application code copying, and also configuring the environment of the image obtained in the outcome.

Containers are considered to be crucial for executing the Cloud Native apps since utilizing them guides the developments towards the deployment of various core Cloud Native principles. As required, a provided Docker container:

- Executes some support logic.
- It is portable around cloud providers provided that it has necessary resources as well as Docker Runtime.
- Clearly declares the software dependencies in DockerFile.
- Installs rapidly for replacing the failed execution of container.

- Duplicates easily for accommodating surplus load on heavily requested functions via launching additional instances of containers.

Deploying CollabNow Microservices

For illustrating the workflow of container-based progress, CollabNow's WebUI microservice has been presented below.

Develop and Push

The CollabNow_web-based DockerFile initially sources the Node.js file from the Docker registry or Docker Hub. Additionally, the Steps of DockerFile entail app configuration, source code file copying, and then guiding Docker about the command to execute the time to launch the app container.

Build & Test

This push activates the consistent integration pipeline build. This integration server constructs the Node.js libraries, and dependencies and then changes the sources into the Docker video. Once constructed, the pipeline constructs this video as a container and further goes ahead to run the unit as well as integration tests. On passing all tests, validated Docker video is pushed to the video registry of the container, where the abstraction layer usually builds on top of object storage, for video tagging, organization of container, and proficient reuse of the current video layers. Company-specific and private container video registries enable the employees to share as well as publish the video to a secure, and central location, however, the public registries enable the popular open source programs and projects such as Node.js for constructing the updated version of software accessible to the developers in already constructed video form.

Deploy & Execute

As CollabNow web UI container video comprising of bug fix has been permitted and accepted and also pushed to the registry, the container orchestration mechanism like Kubernetes can "pull" this video and install it over the host having Docker runtime. A narrow view of accessible resources for the executing container, specified by the developer is done and is run from a different container in isolation. A single host having container runtime deployed can execute distinct containers at a time, depending on the resource availability.

CLUSTERS

Kubernetes are organized as clusters of virtual machines. These machines are known as the nodes that share the computing, storage as well as network resources. Each cluster has a master node that is linked to more than one worker node. These are in charge of executing the sets of containerization workloads and applications, which are known as pods, where the master node maintains and manages the pods that need to be executed on the respective worker nodes.

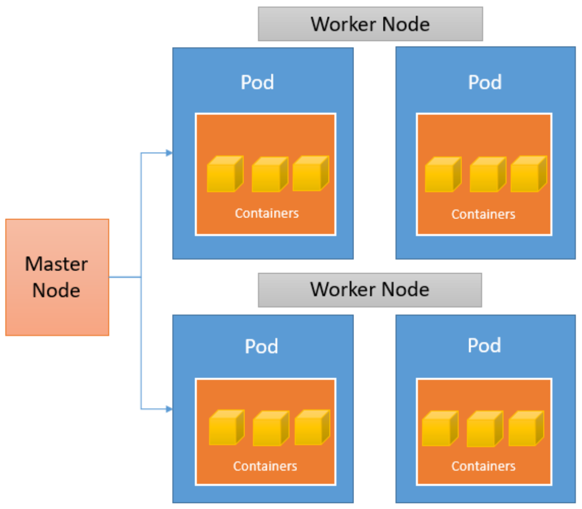


Figure 1. Kubernetes Cluster [1]

KUBERNETES

Kubernetes container orchestration mechanism was first initiated by Google with the help of engineers who assisted in architecting and developing an internal cluster of Google. The team of engineers built the open source container management system (CMS) integrating several lessons learned from designing an internal container platform. Since then, Kubernetes has matured enough quickly and has been rolled out in various big production installations. In March 2018, It became

1st project to graduate from “Cloud Native Computing Foundation”, representing that it has turned out to be mature and also stable enough to tackle large-scale deployments while acquiring and attaining high-level security as well as code quality.

Design Overview

Kubernetes are known as the container cluster having a dynamic system that is capable to handle the deployments and interconnection of these containers on a group of worker servers. The servers (workers) where containers execute are known as Nodes and servers that manage all of these executions of the containers are known as the control panel of Kubernetes.

Kubernetes Deployments

Pods are usually subdivided by making use of the Deployments, known as the objects defining YAML files declaring the specifically desired state. Once it is built, the controller on Control Plane slowly brings the original cluster state to match the required state mentioned in Deployment by scheduling the containers over the Nodes that are desired. Through Deployments, a service owner can conveniently scale a set of Pod replicas in a horizontal direction or also by performing the zero-downtime scaling update into a new container video version by just editing the YAML file and then performing the API call. However, these deployments can be paused, rolled back, or resumed quickly at any time.

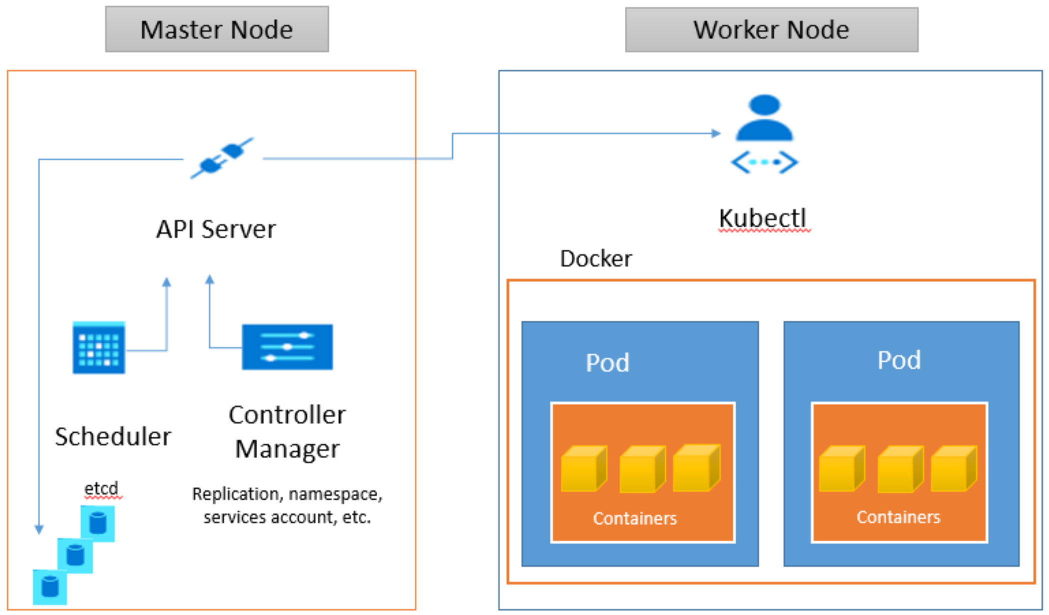


Figure 2. Kubernetes Deployment *[2]*

Kubernetes Services

Once the Kubernetes are deployed, the Service is initiated to enable groups of alike Pods for receiving the traffic. These services are utilized for granting a group of Pods replicas a static IP and then configure the load balancing among them through utilizing the cloud provider or user-specified custom load Balancer. Services also enable users to leverage the cloud provider-based firewalls to lock down external access.

Kubernetes Node Agents

To initiate as well as manage the pods along with the containers on machines, Nodes execute the agent procedure known as Kubelet that communicates with that of the Kube-apiserver over the Control Plane. Making use of the container runtime such as Docker, it is also executed on Nodes, these scheduled containers are at first pulled as the video from public or private registry, and it is then created, after which it is launched. Nodes execute the Kube-proxy as well, which manages the rules of the network on the host.

Kubernetes – Control Plane

This segment runs the Nodes and also manages its scheduling and performs maintenance of its workloads. It comprises of Kube-Apiserver front-end, which is backed through “key-value store - etcd” for storing cluster data. At the final stage, Kube-scheduler programs all the Pods to the Nodes, and a set of controllers uninterruptedly observe the cluster state and drive the original state towards the desired state.

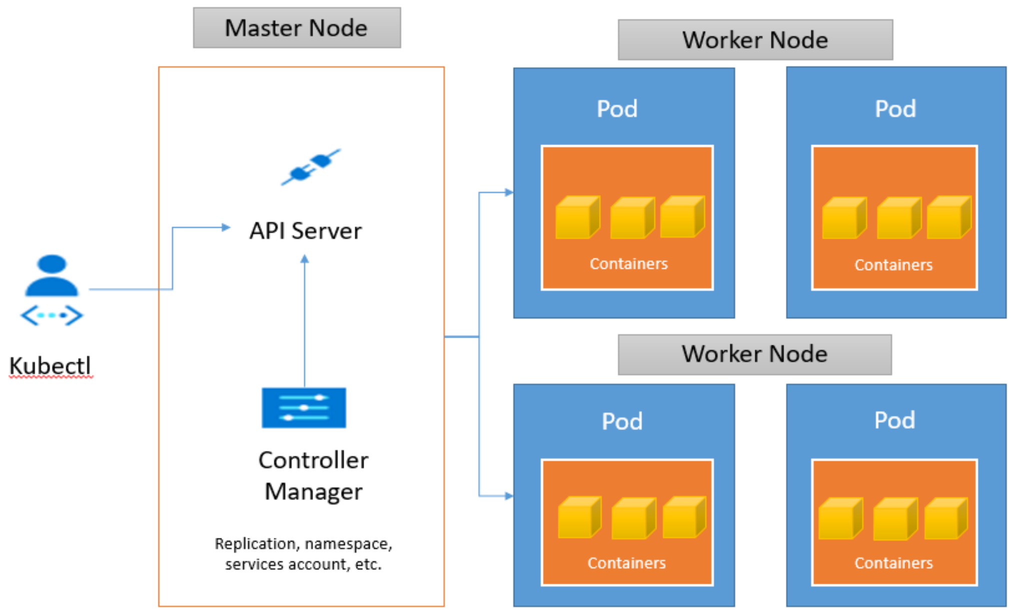


Figure 3. Kubernetes Control Plane [1]

CollabNow video messaging works on Kubernetes

Kubernetes for CollabNow video messaging delivers a simple as well as cost-effective solution for developers who seek to install and execute the containerized CollabNow video messaging feature on the Kubernetes cluster. Using Kubernetes, it becomes convenient for developers to rapidly launch the CollabNow video messaging feature onto the managed as well as self-healing environment of

Kubernetes without any sort of provisioning, installation, or managing the cluster from scratch. Rolling over the cluster of Kubernetes usually entails various steps that are as follows:

- Provisioning of Underlying computation, networking, and storage infrastructure for Nodes as well as control plane;
- Kubernetes configuration and bootstrapping such as etcd cluster or Pod Networking;
- Testing cluster thoroughly for resiliency towards failures of infrastructure

As soon as the setup completes, Kubernetes clusters are changed so that they can be monitored as well managed by DevOps teams, however, daily maintenance tasks such as cluster upgradation needs manual interference by engineers.

We here at CollabNow have assessed the Kubernetes and have rolled out a microservices-based video messaging feature on the platform. Implemented on Kubernetes, CollabNow comprises the following 3 pods (microservices).

- Web UI
- Video Management
- User Management

Once the desired Node number is provided to the cluster, CollabNow's developers generate the Kubernetes Deployment as well as services for every microservice element, referencing the Docker video through the private registry. Making use of

the Kubernetes secrets, they also become capable of storing the database information along with other crucial credentials such as Pods having gain to access to the environment.

In the same way, they could make use of the Kubectl along with the preferred Kubernetes management tool for launching the services into the cluster. As soon as it gets launched, Kubernetes automatically configure the remaining cloud infrastructure like Load Balancer, Block Storage as well as Firewall, as per the developer.

After that, service owners can independently scale the services initially by increasing the number of replica Pods in the “Deployment Configuration File” and launching deployment through Kubernetes API. This similar flow is automated and also combined with CD/CI pipeline for continuously rolling out new application versions.

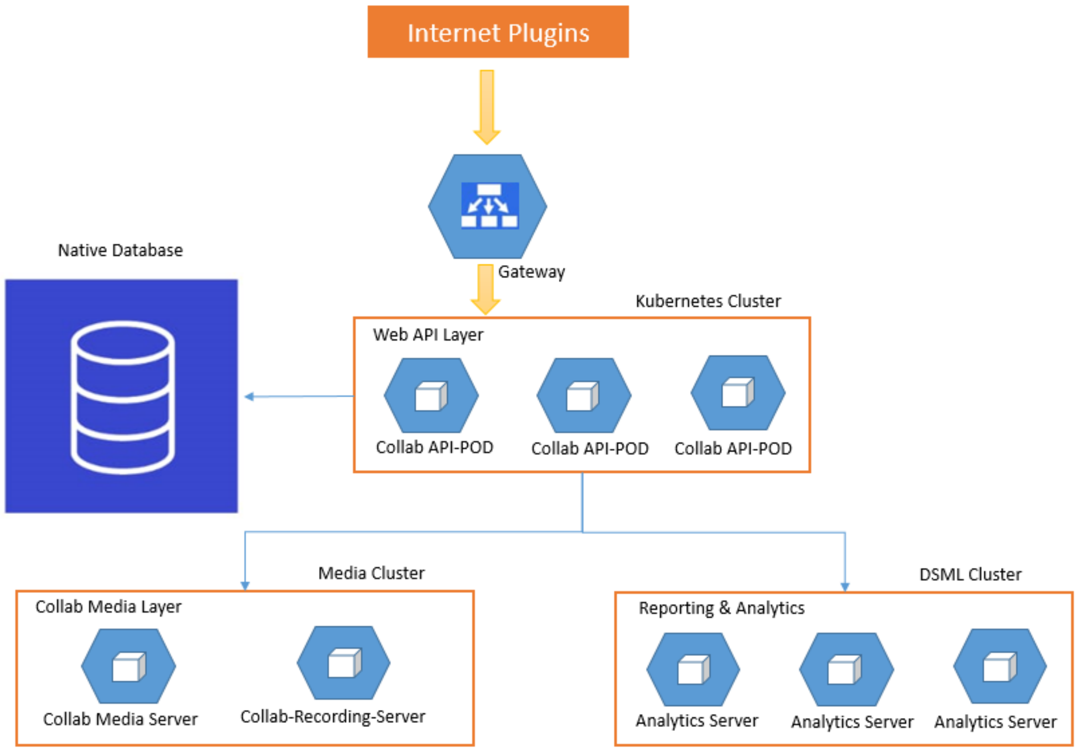


Figure 4. CollabNow Kuberentes Architecture

FINAL WORDS

CollabNow is a video collaboration platform that provides a scalable productive and engaging collaboration to the user from any location and any device. Our business is focused on improving the productivity and efficiency of remote and hybrid teams by providing a natural collaboration experience For those in IT who are responsible for provisioning and managing video conferencing systems, this application represents an opportunity to get ahead of the coming wave in video adoption. CollabNow's distributed and microservices architecture with enterprise-level broader security provides a better experience for video, audio, and content to accelerate productivity, collaboration, and innovation. CollabNow has built a unified communication solution that can enhance productivity, and innovation using proprietary video collaboration and insights platform.

CollabNow has introduced a new feature of video messaging on the Kubernetes that provides simple and cost-effective solution for developers who seek to run and execute the containerized CollabNow video messaging feature on the Kubernetes cluster.

With the help of Kubernetes deployment, CollabNow has made it convenient for the developers to rapidly launch the video messaging feature onto the managed and self-healing environment of Kubernetes cluster without any sort of provisioning, installation, or managing the cluster from scratch.

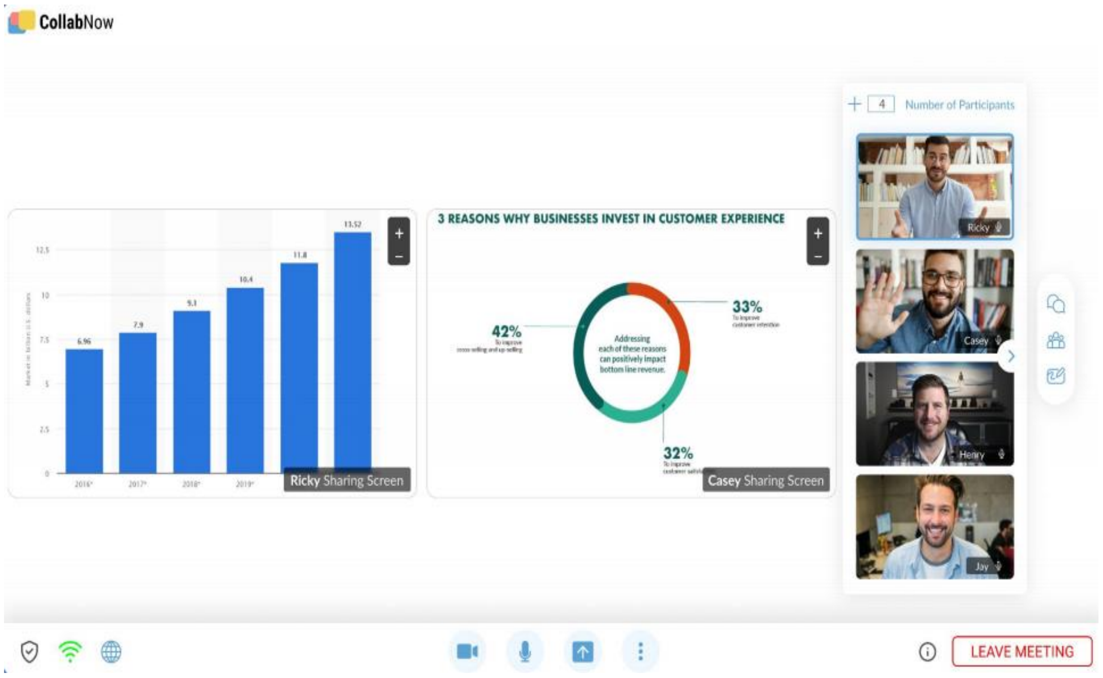
REFERENCES

[1] "Kubernetes: Getting Started | Microsoft Azure." https://azure.microsoft.com/en-us/solutions/kubernetes-on-azure/get-started/ (accessed Aug. 02, 2022).

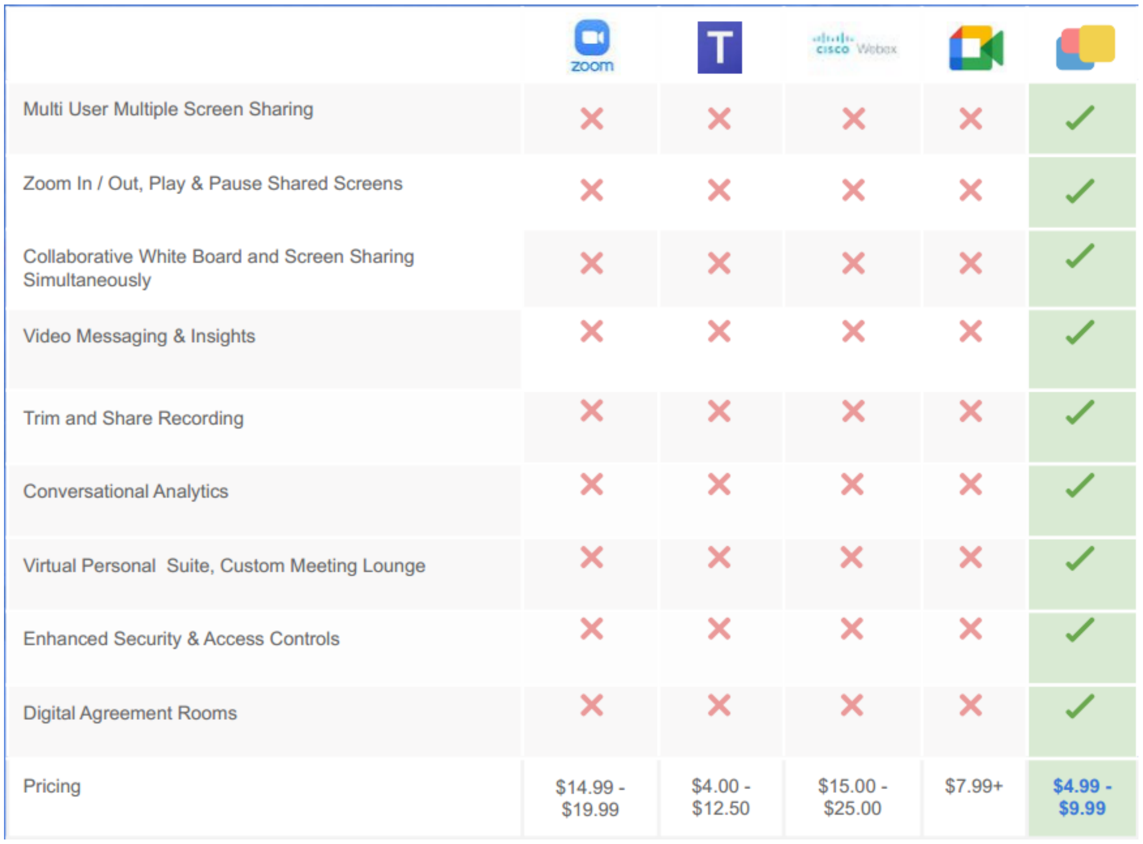
[2] J. Petazzoni, "Kubernetes Deployments: The Ultimate Guide," *Semaphore*, Jul. 16, 2019. https://semaphoreci.com/blog/kubernetes-deployment (accessed Aug. 02, 2022).

APPENDICES

Appendix 1 – Multiuser Multi Screen – Rich Collaboration


CollabNow
4 Number of Participants
3 REASONS WHY BUSINESSES INVEST IN CUSTOMER EXPERIENCE
42%
33%
32%
Addressing each of these reasons can positively impact bottom line revenue.
Ricky Sharing Screen
Casey Sharing Screen
Ricky
Casey
Henry
Jay
LEAVE MEETING

Appendix 2 – Competitor Matrix



	zoom	T	cisco Webex		
Multi User Multiple Screen Sharing	✗	✗	✗	✗	✓
Zoom In / Out, Play & Pause Shared Screens	✗	✗	✗	✗	✓
Collaborative White Board and Screen Sharing Simultaneously	✗	✗	✗	✗	✓
Video Messaging & Insights	✗	✗	✗	✗	✓
Trim and Share Recording	✗	✗	✗	✗	✓
Conversational Analytics	✗	✗	✗	✗	✓
Virtual Personal Suite, Custom Meeting Lounge	✗	✗	✗	✗	✓
Enhanced Security & Access Controls	✗	✗	✗	✗	✓
Digital Agreement Rooms	✗	✗	✗	✗	✓
Pricing	$14.99 - $19.99	$4.00 - $12.50	$15.00 - $25.00	$7.99+	**$4.99 - $9.99**

CollabNow - Delight Remote Collaboration

Corporate Headquarter

1318 Knolls Creek Dr

Danville, CS USA 94506

925.905.9066

www.collabnow.ai

info@collabnow.ai