Project Report

Build a CI/CD Pipeline on Kubernetes with Argo Rollouts, ArgoCD, Argo Workflows and Events, Argo Image Updater

Learning Report: Ultimate Argo Bootcamp by Gourav J. Shah

Course Overview

Course Title: *Ultimate Argo Bootcamp – From Zero to Production-Ready CI/CD Pipelines*
Instructor: Gourav J. Shah, Founder – *School of DevOps™*
Specialization: DevOps | AIOps | MLOps | GitOps Automation
Course Type: Hands-on, Real-World Bootcamp with Practical Use Case

---

Introduction

The **Ultimate Argo Bootcamp** by *Gourav J. Shah* is not just another online course — it's a **complete real-world learning experience** built around **hands-on implementation, troubleshooting, and automation**.

Throughout this bootcamp, I worked step by step on a **real production-grade use case**, covering the full lifecycle of **Continuous Integration and Continuous Delivery (CI/CD)** using the **Argo suite of tools** — including ArgoCD, Argo Rollouts, Argo Workflows, Argo Events, and Argo Image Updater.

This course focused heavily on **practical learning and problem-solving**, where I encountered and resolved real-world issues like synchronization errors, misconfigured manifests, rollout failures, and namespace conflicts. These troubleshooting experiences helped me understand how DevOps pipelines behave in live environments and how to identify, analyze, and fix issues effectively.

Unlike other theoretical programs, this bootcamp pushed me to **learn by doing**. Each module represented a milestone in building a complete CI/CD ecosystem — from **source code commits to automated deployment, monitoring, and rollback**. I captured **screenshots for every step** of my implementation and testing process, not just to document progress but to ensure reproducibility and smoother execution in future projects.

The bootcamp helped me gain strong control over **real-world DevOps pipelines**, including setting up GitOps workflows, debugging Argo sync issues, implementing Blue/Green and Canary rollouts, integrating monitoring with Prometheus and Grafana, and automating everything through Argo Events.

By the end, I didn't just "learn" Argo — I **built, tested, troubleshooted, and optimized** a fully functional, production-ready CI/CD system from scratch.

What I Learned

Through the **Ultimate Argo Bootcamp**, I acquired deep technical knowledge and hands-on experience in automating end-to-end software delivery pipelines.

1. Building a CI/CD Pipeline Using Argo Suite

- Designed a complete multi-stage CI/CD pipeline integrating:
  - **ArgoCD** for continuous delivery with GitOps
  - **Argo Rollouts** for progressive deployments
  - **Argo Workflows** for build/test automation
  - **Argo Events** for trigger-based automation
  - **Argo Image Updater** for dynamic image updates

2. Advanced Deployment Strategies

- Implemented **Blue/Green** and **Canary deployments** using Argo Rollouts.
- Learned to perform controlled rollouts with **Nginx Ingress Controller** for traffic shifting.
- Configured **automatic rollback** on failure analysis for stable production environments.

3. GitOps and Continuous Delivery

- Applied **GitOps principles** to automatically sync Git repositories with Kubernetes clusters.
- Managed real-time changes through pull requests, commits, and declarative manifests using ArgoCD.

4. Workflow Automation

- Created **DAG (Directed Acyclic Graph)**-based pipelines using Argo Workflows.
- Automated build, test, and deploy stages for microservices applications on Kubernetes.

5. Event-Driven Pipelines

- Configured **Argo Events** to trigger deployments and workflows automatically on Git updates.
- Implemented event-based notifications for seamless pipeline automation.

6. Observability and Monitoring

- Integrated **Prometheus** and **Grafana** dashboards to monitor cluster and application health.
- Configured **automated analysis** for rollout validation and rollback using metric thresholds.

Real-World Troubleshooting Experience

During this course, I faced and fixed multiple real-world issues such as:

- ArgoCD **OutOfSync** and **HealthStatus** problems
- Kubernetes deployment version mismatches
- YAML syntax and API schema errors
- **Failed Rollouts** during Canary testing
- Event trigger misconfigurations and permission issues

Each challenge strengthened my debugging mindset and helped me build confidence in maintaining **production-grade reliability**.

Key Takeaways

✅ Built and deployed a **production-ready CI/CD pipeline** using the complete Argo ecosystem.
✅ Understood and implemented **progressive delivery strategies** with automation and monitoring.
✅ Learned to **troubleshoot and optimize** real-world DevOps systems.
✅ Acquired strong expertise in **GitOps principles, Kubernetes deployment, and observability tools**.
✅ Developed a full documentation trail with **screenshots for every configuration and fix**, ensuring reproducibility.

Conclusion

The **Ultimate Argo Bootcamp** was one of the most intensive and valuable DevOps learning experiences I've completed. It gave me **real-world exposure** to end-to-end automation — from CI/CD setup to monitoring, event triggers, and advanced deployment strategies.

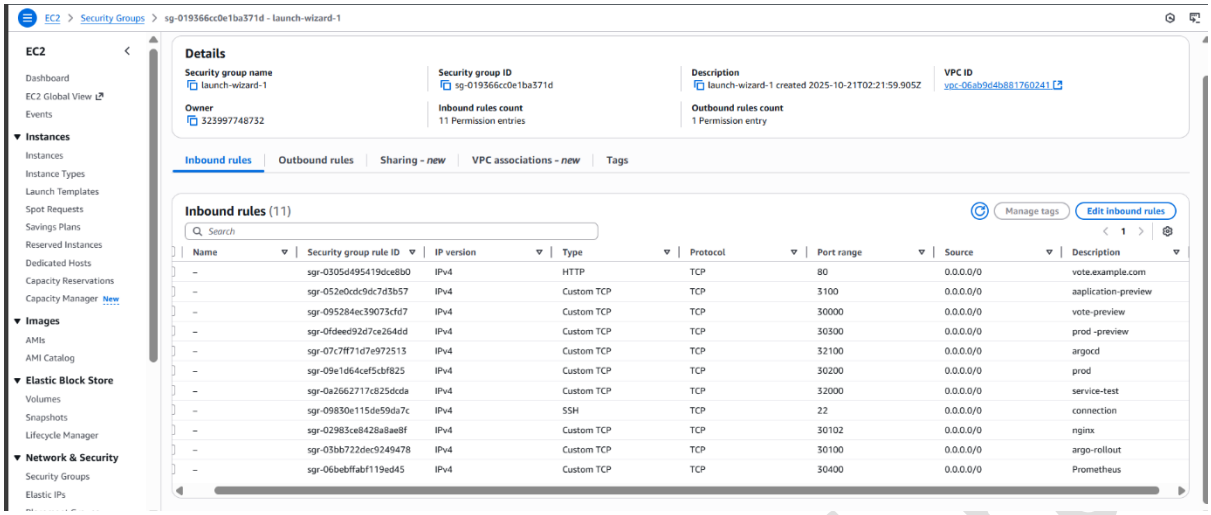
By capturing every step through detailed screenshots and notes, I not only ensured smoother project execution but also built a ready reference for future production projects.

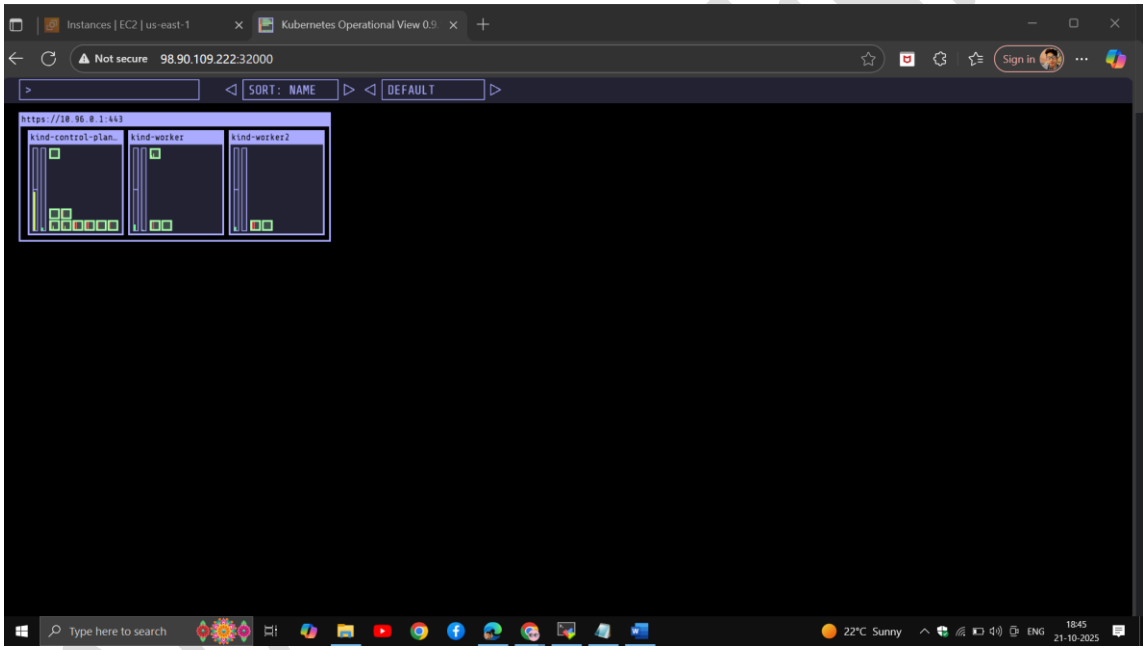
This course truly enhanced my ability to **plan, build, deploy, and troubleshoot real DevOps pipelines**, making me confident and industry-ready for any complex CI/CD environment.

(All related screenshots and workflow documentation are attached below to demonstrate the complete hands-on implementation and troubleshooting process.)

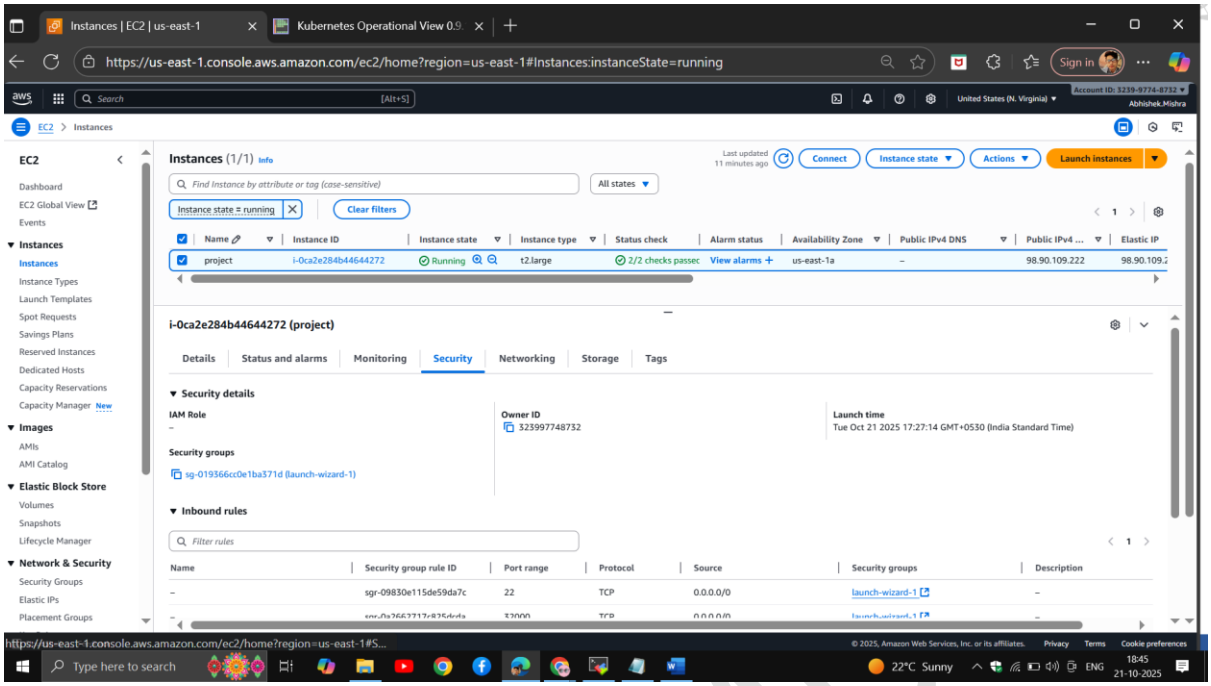
GitHub Repo: https://github.com/Abhi-mishra998/argo-labs.git

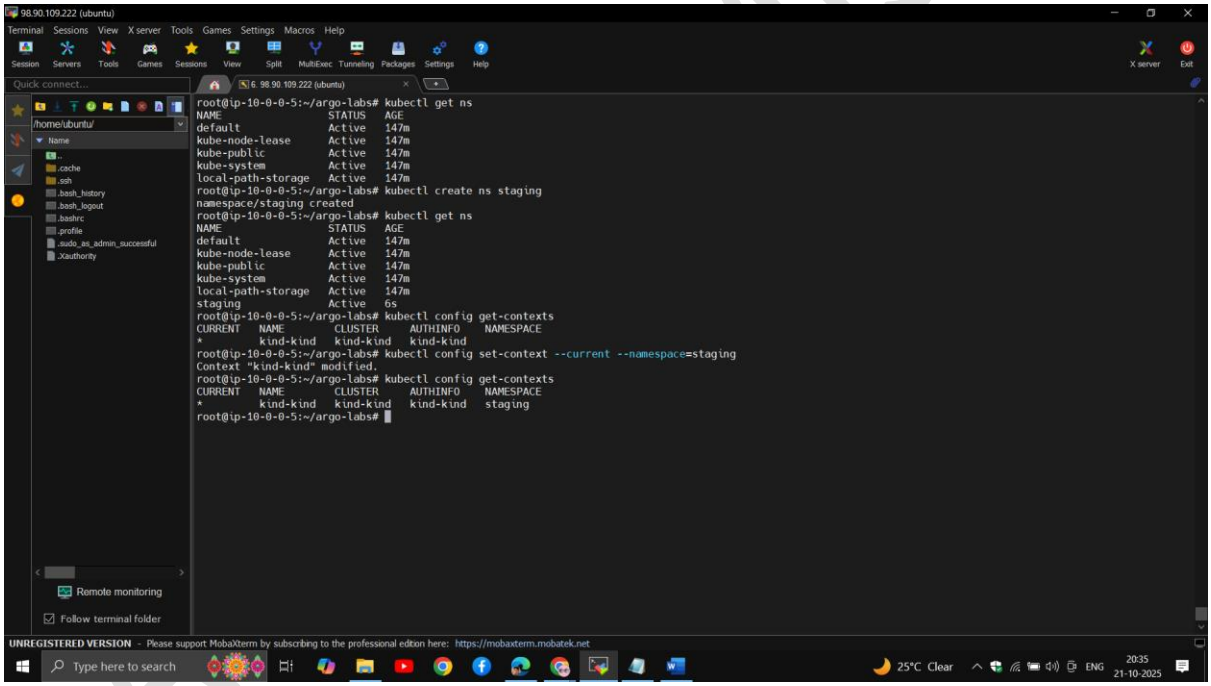
Whenever I deployed or exposed a service, I made sure to specify the port numbers clearly and document them using AWS-style descriptions.
This helped me remember which service was running on which port and ensured consistency across my deployments.
By following this approach, I could easily identify and troubleshoot services during testing and scaling.
It also improved clarity while configuring load balancers, security groups, and ingress rules in AWS.
Maintaining a proper port-to-service mapping became an essential practice for smoother and more reliable DevOps operations.

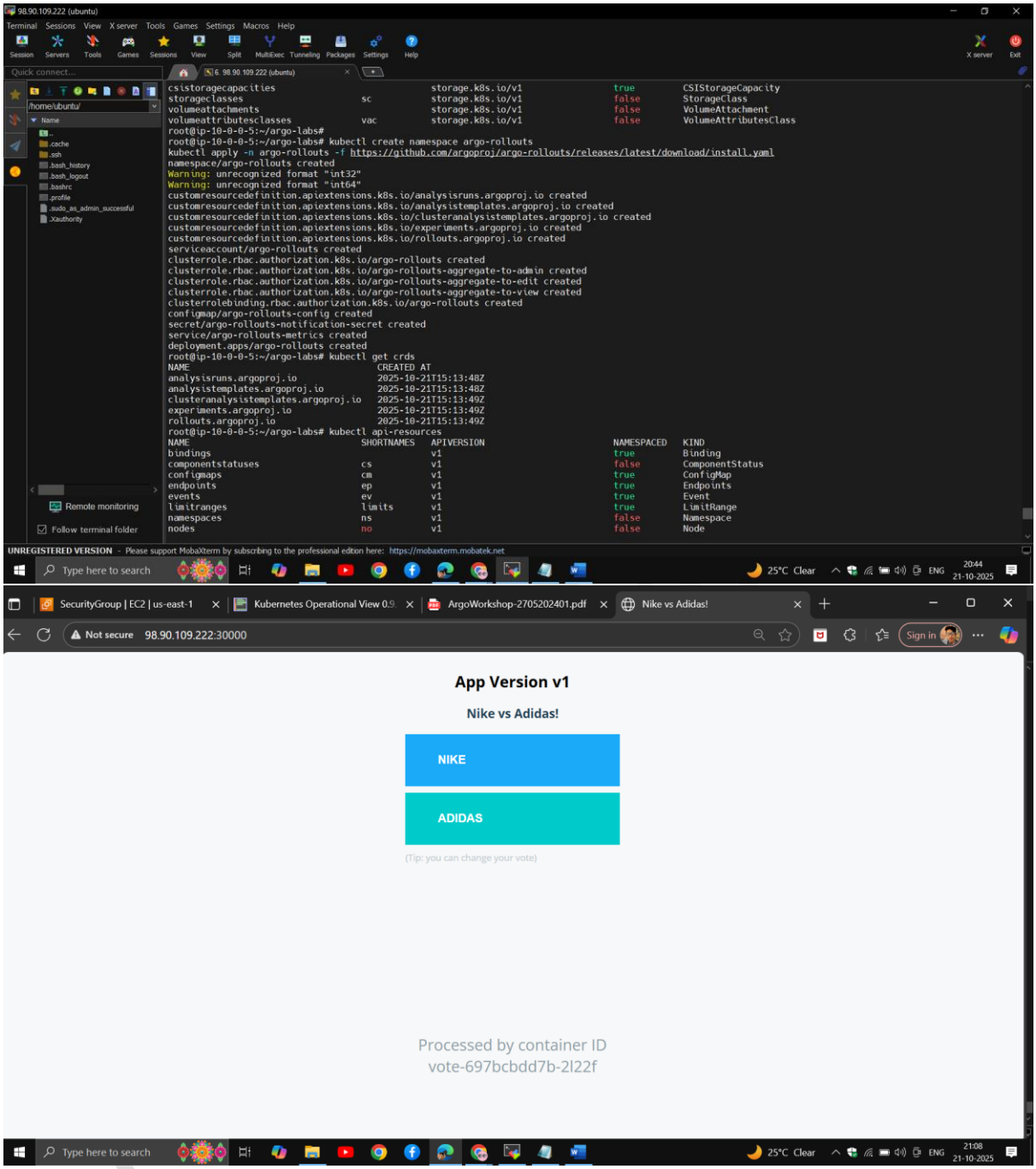

EC2 > Security Groups > sg-019366cc0e1ba371d - launch-wizard-1
EC2
Dashboard
EC2 Global View
Events
Instances
Instances
Instance Types
Launch Templates
Spot Requests
Savings Plans
Reserved Instances
Dedicated Hosts
Capacity Reservations
Capacity Manager New
Images
AMIs
AMI Catalog
Elastic Block Store
Volumes
Snapshots
Lifecycle Manager
Network & Security
Security Groups
Elastic IPs
Details
Security group name
launch-wizard-1
Security group ID
sg-019366cc0e1ba371d
Description
launch-wizard-1 created 2025-10-21T02:21:59.905Z
VPC ID
vpc-06ab9d4b881760241
Owner
323997748732
Inbound rules count
11 Permission entries
Outbound rules count
1 Permission entry
Inbound rules
Outbound rules
Sharing - new
VPC associations - new
Tags
Inbound rules (11)
Search
Manage tags
Edit inbound rules
Name
Security group rule ID
IP version
Type
Protocol
Port range
Source
Description
sgr-0305d495419dce8b0 IPv4 HTTP TCP 80 0.0.0.0/0 vote.example.com
sgr-052e0cdc9dc7d3b57 IPv4 Custom TCP TCP 3100 0.0.0.0/0 aaplication-preview
sgr-095284ec39073cfd7 IPv4 Custom TCP TCP 30000 0.0.0.0/0 vote-preview
sgr-0fdeed92d7ce264dd IPv4 Custom TCP TCP 30300 0.0.0.0/0 prod -preview
sgr-07c7ff71d7e972513 IPv4 Custom TCP TCP 32100 0.0.0.0/0 argocd
sgr-09e1d64cef5cbf825 IPv4 Custom TCP TCP 30200 0.0.0.0/0 prod
sgr-0a2662717c825dcda IPv4 Custom TCP TCP 32000 0.0.0.0/0 service-test
sgr-09830e115de59da7c IPv4 SSH TCP 22 0.0.0.0/0 connection
sgr-02983ce8428a8ae8f IPv4 Custom TCP TCP 30102 0.0.0.0/0 nginx
sgr-03bb722dec9249478 IPv4 Custom TCP TCP 30100 0.0.0.0/0 argo-rollout
sgr-06bebffabf119ed45 IPv4 Custom TCP TCP 30400 0.0.0.0/0 Prometheus

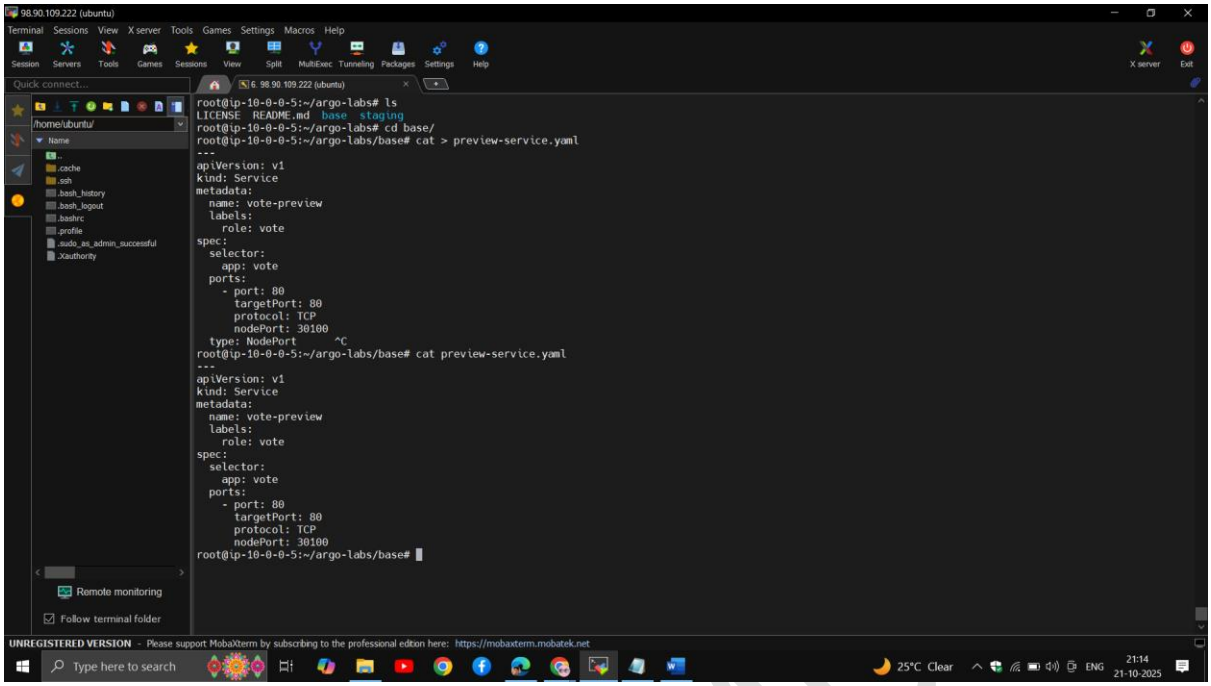

Instances | EC2 | us-east-1
Kubernetes Operational View 0.9
Not secure 98.90.109.222:32000
Sign in
SORT: NAME
DEFAULT
https://10.96.0.1:443
kind-control-plan...
kind-worker
kind-worker2
Type here to search
22°C Sunny
ENG
18:45
21-10-2025

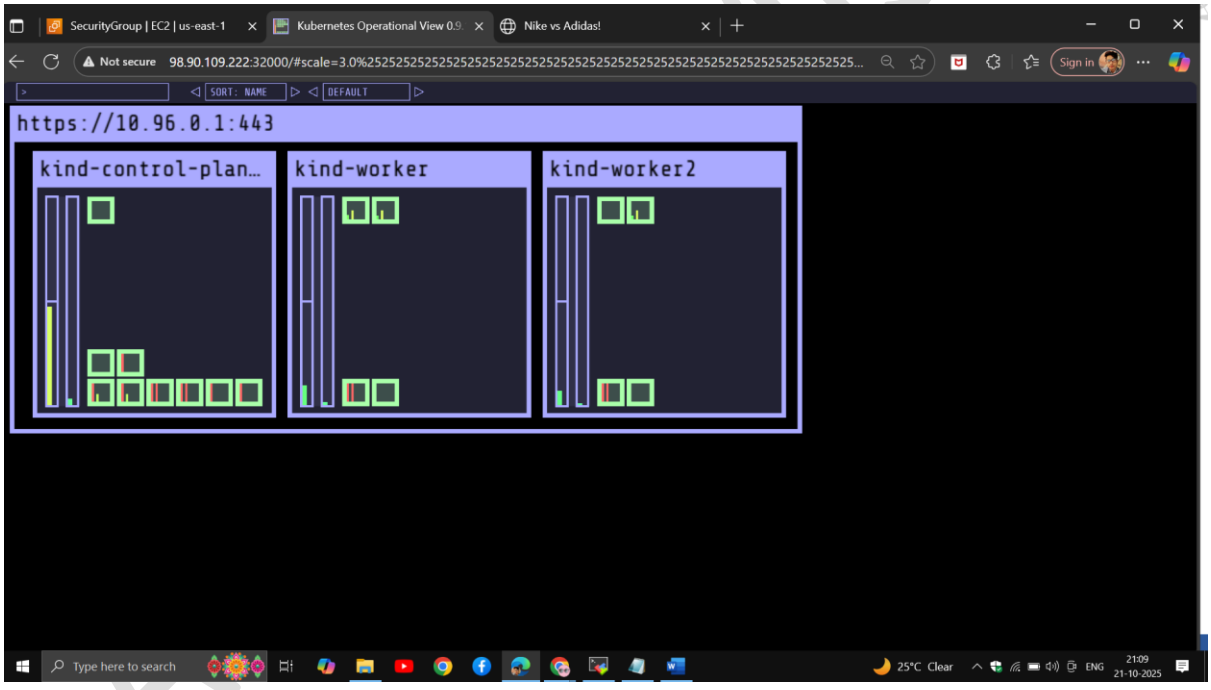

Instances | EC2 | us-east-1
Kubernetes Operational View 0.9
Not secure 98.90.109.222:32000/#scale=3.0%252525252525252F
SORT: NAME
DEFAULT
ttps://10.96.0.1:443
kind-control-plan...
kind-worker
kind-worker2
Type here to search
22°C Sunny
ENG
18:51
21-10-2025

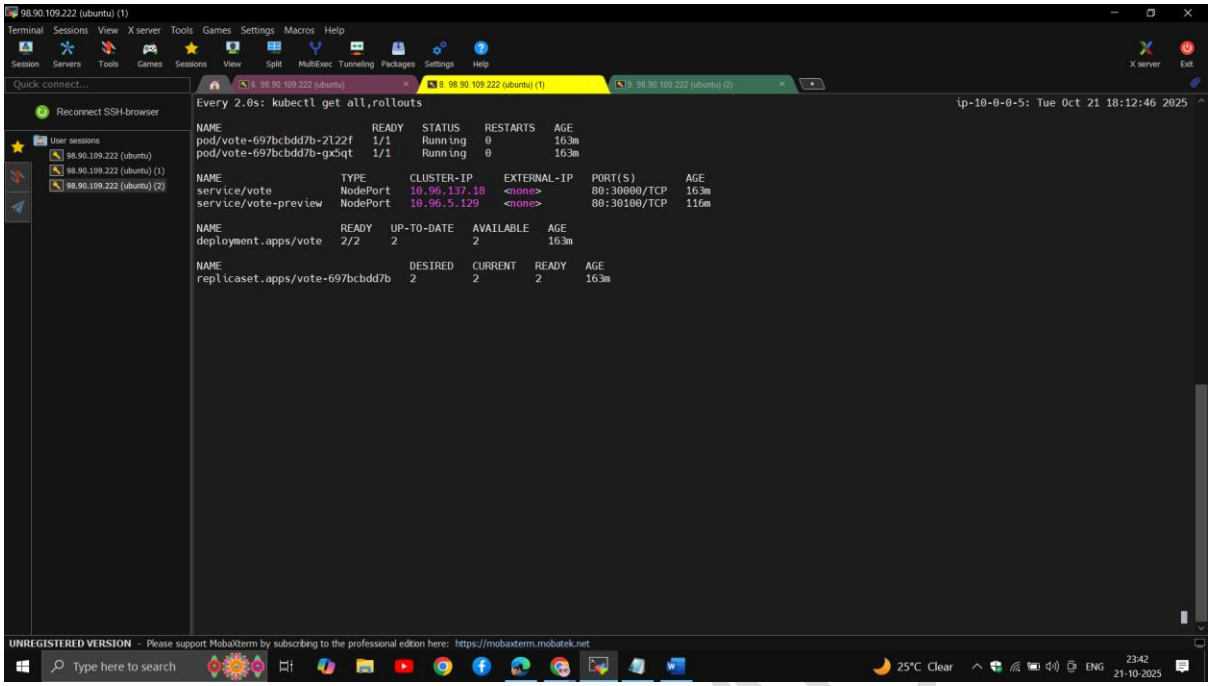

Instances | EC2 | us-east-1
Kubernetes Operational View 0.9
https://us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#Instances:instanceState=running
EC2 > Instances
Instances (1/1) Info
Connect
Instance state
Actions
Launch instances
Instance state = running
Clear filters
project
i-0ca2e284b44644272
Running
t2.large
2/2 checks passed
View alarms
us-east-1a
98.90.109.222
i-0ca2e284b44644272 (project)
Details
Status and alarms
Monitoring
Security
Networking
Storage
Tags
Security details
IAM Role
Owner ID
323997748732
Launch time
Tue Oct 21 2025 17:27:14 GMT+0530 (India Standard Time)
Security groups
sg-019366cc0e1ba371d (launch-wizard-1)
Inbound rules
Name
Security group rule ID
Port range
Protocol
Source
Security groups
Description
sgr-09830e115de59da7c
22
TCP
0.0.0.0/0
launch-wizard-1
Type here to search
22°C Sunny
ENG
18:45
21-10-2025

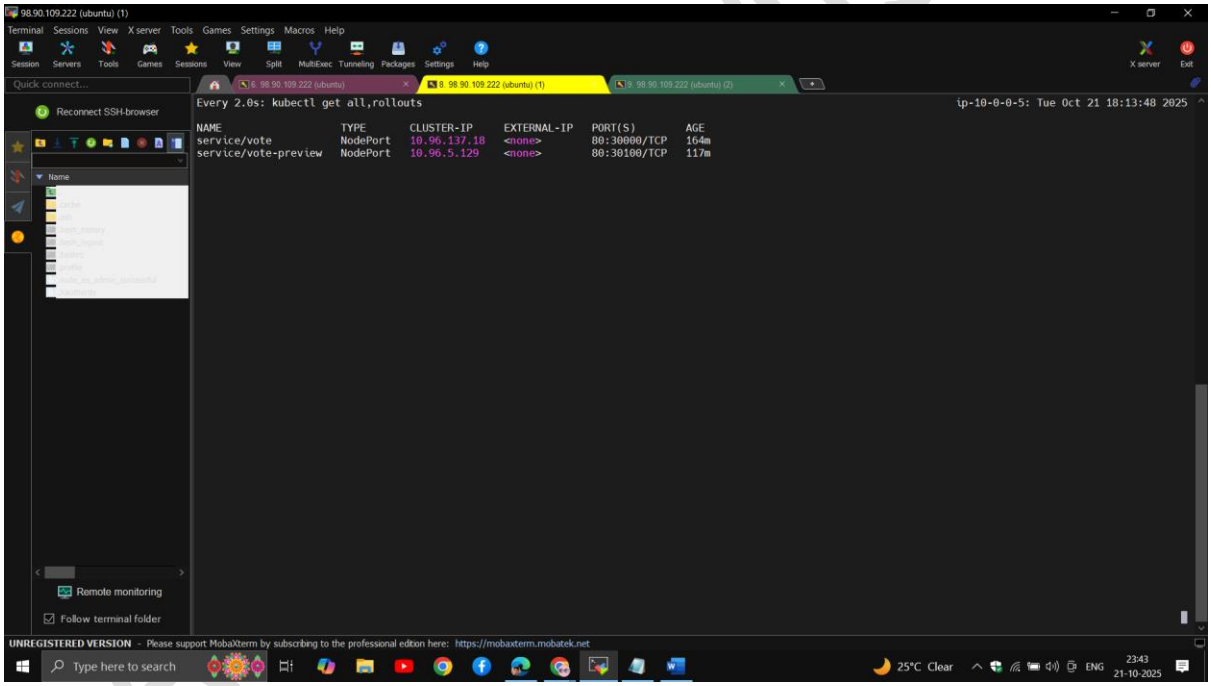

98.90.109.222 (ubuntu)
root@ip-10-0-0-5:~/argo-labs# kubectl get ns
NAME STATUS AGE
default Active 147m
kube-node-lease Active 147m
kube-public Active 147m
kube-system Active 147m
local-path-storage Active 147m
root@ip-10-0-0-5:~/argo-labs# kubectl create ns staging
namespace/staging created
root@ip-10-0-0-5:~/argo-labs# kubectl get ns
NAME STATUS AGE
default Active 147m
kube-node-lease Active 147m
kube-public Active 147m
kube-system Active 147m
local-path-storage Active 147m
staging Active 6s
root@ip-10-0-0-5:~/argo-labs# kubectl config get-contexts
CURRENT NAME CLUSTER AUTHINFO NAMESPACE
* kind-kind kind-kind kind-kind
root@ip-10-0-0-5:~/argo-labs# kubectl config set-context --current --namespace=staging
Context "kind-kind" modified.
root@ip-10-0-0-5:~/argo-labs# kubectl config get-contexts
CURRENT NAME CLUSTER AUTHINFO NAMESPACE
* kind-kind kind-kind kind-kind staging
root@ip-10-0-0-5:~/argo-labs#
Remote monitoring
Follow terminal folder
UNREGISTERED VERSION - Please support MobaXterm by subscribing to the professional edition here: https://mobaxterm.mobatek.net
Type here to search
25°C Clear
ENG
20:35
21-10-2025

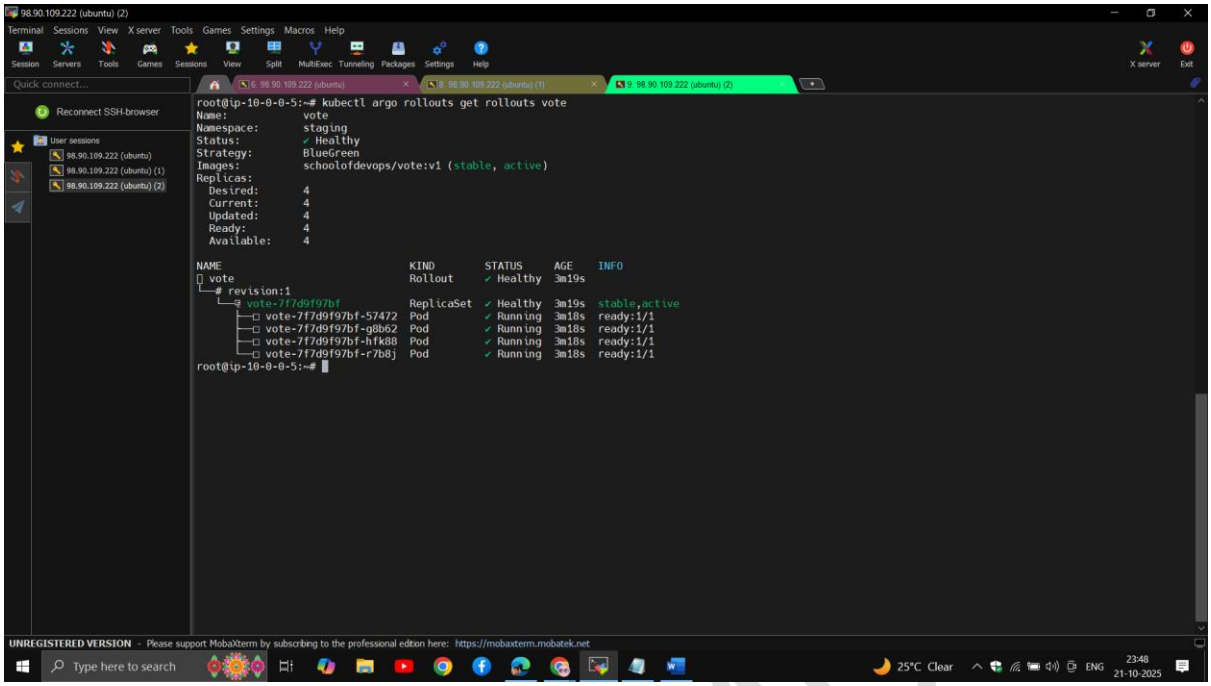

98.90.109.222 (ubuntu)
csistoragecapacities storage.k8s.io/v1 true CSIStorageCapacity
storageclasses sc storage.k8s.io/v1 false StorageClass
volumeattachments storage.k8s.io/v1 false VolumeAttachment
volumeattributesclasses vac storage.k8s.io/v1 false VolumeAttributesClass
root@ip-10-0-0-5:~/argo-labs#
root@ip-10-0-0-5:~/argo-labs# kubectl create namespace argo-rollouts
kubectl apply -n argo-rollouts -f https://github.com/argoproj/argo-rollouts/releases/latest/download/install.yaml
namespace/argo-rollouts created
Warning: unrecognized format "int32"
Warning: unrecognized format "int64"
customresourcedefinition.apiextensions.k8s.io/analysisruns.argoproj.io created
customresourcedefinition.apiextensions.k8s.io/analysistemplates.argoproj.io created
customresourcedefinition.apiextensions.k8s.io/clusteranalysistemplates.argoproj.io created
customresourcedefinition.apiextensions.k8s.io/experiments.argoproj.io created
customresourcedefinition.apiextensions.k8s.io/rollouts.argoproj.io created
serviceaccount/argo-rollouts created
clusterrole.rbac.authorization.k8s.io/argo-rollouts created
clusterrole.rbac.authorization.k8s.io/argo-rollouts-aggregate-to-admin created
clusterrole.rbac.authorization.k8s.io/argo-rollouts-aggregate-to-edit created
clusterrole.rbac.authorization.k8s.io/argo-rollouts-aggregate-to-view created
clusterrolebinding.rbac.authorization.k8s.io/argo-rollouts created
configmap/argo-rollouts-config created
secret/argo-rollouts-notification-secret created
service/argo-rollouts-metrics created
deployment.apps/argo-rollouts created
root@ip-10-0-0-5:~/argo-labs# kubectl get crds
NAME CREATED AT
analysisruns.argoproj.io 2025-10-21T15:13:48Z
analysistemplates.argoproj.io 2025-10-21T15:13:48Z
clusteranalysistemplates.argoproj.io 2025-10-21T15:13:49Z
experiments.argoproj.io 2025-10-21T15:13:49Z
rollouts.argoproj.io 2025-10-21T15:13:49Z
root@ip-10-0-0-5:~/argo-labs# kubectl api-resources
NAME SHORTNAMES APIVERSION NAMESPACED KIND
bindings v1 true Binding
componentstatuses cs v1 false ComponentStatus
configmaps cm v1 true ConfigMap
endpoints ep v1 true Endpoints
events ev v1 true Event
limitranges limits v1 true LimitRange
namespaces ns v1 false Namespace
nodes no v1 false Node
Remote monitoring
Follow terminal folder
UNREGISTERED VERSION - Please support MobaXterm by subscribing to the professional edition here: https://mobaxterm.mobatek.net
Type here to search
25°C Clear
ENG
20:44
21-10-2025
SecurityGroup | EC2 | us-east-1
Kubernetes Operational View 0.9
ArgoWorkshop-2705202401.pdf
Nike vs Adidas!
Not secure 98.90.109.222:30000
Sign in
App Version v1
Nike vs Adidas!
NIKE
ADIDAS
(Tip: you can change your vote)
Processed by container ID
vote-697bcbdd7b-2l22f
Type here to search
25°C Clear
ENG
21:08
21-10-2025

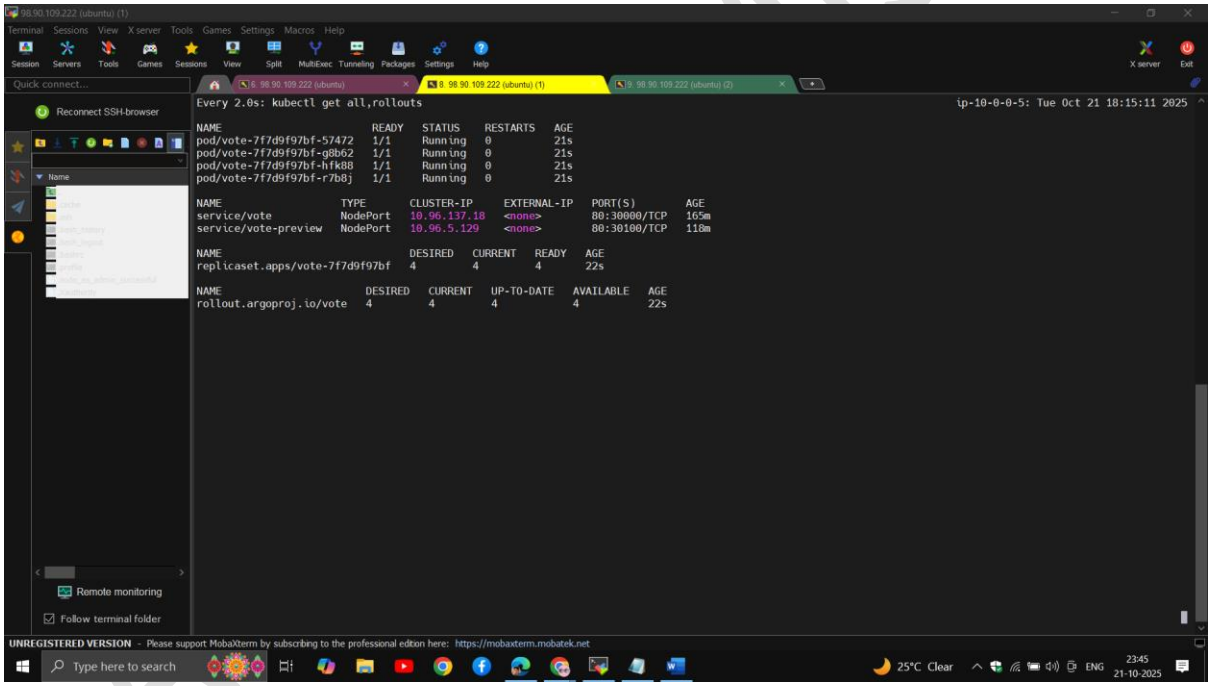

98.90.109.222 (ubuntu)
Terminal Sessions View X server Tools Games Settings Macros Help
Session Servers Tools Games Sessions View Split MultiExec Tunneling Packages Settings Help
X server Exit
Quick connect...
6. 98.90.109.222 (ubuntu)
/home/ubuntu/
Name
.cache
.ssh
.bash_history
.bash_logout
.bashrc
.profile
.sudo_as_admin_successful
.Xauthority
Remote monitoring
Follow terminal folder
root@ip-10-0-0-5:~/argo-labs# ls
LICENSE README.md base staging
root@ip-10-0-0-5:~/argo-labs# cd base/
root@ip-10-0-0-5:~/argo-labs/base# cat > preview-service.yaml
---
apiVersion: v1
kind: Service
metadata:
name: vote-preview
labels:
role: vote
spec:
selector:
app: vote
ports:
- port: 80
targetPort: 80
protocol: TCP
nodePort: 30100
type: NodePort ^C
root@ip-10-0-0-5:~/argo-labs/base# cat preview-service.yaml
---
apiVersion: v1
kind: Service
metadata:
name: vote-preview
labels:
role: vote
spec:
selector:
app: vote
ports:
- port: 80
targetPort: 80
protocol: TCP
nodePort: 30100
root@ip-10-0-0-5:~/argo-labs/base#
UNREGISTERED VERSION - Please support MobaXterm by subscribing to the professional edition here: https://mobaxterm.mobatek.net
Type here to search
25°C Clear
ENG
21:14
21-10-2025

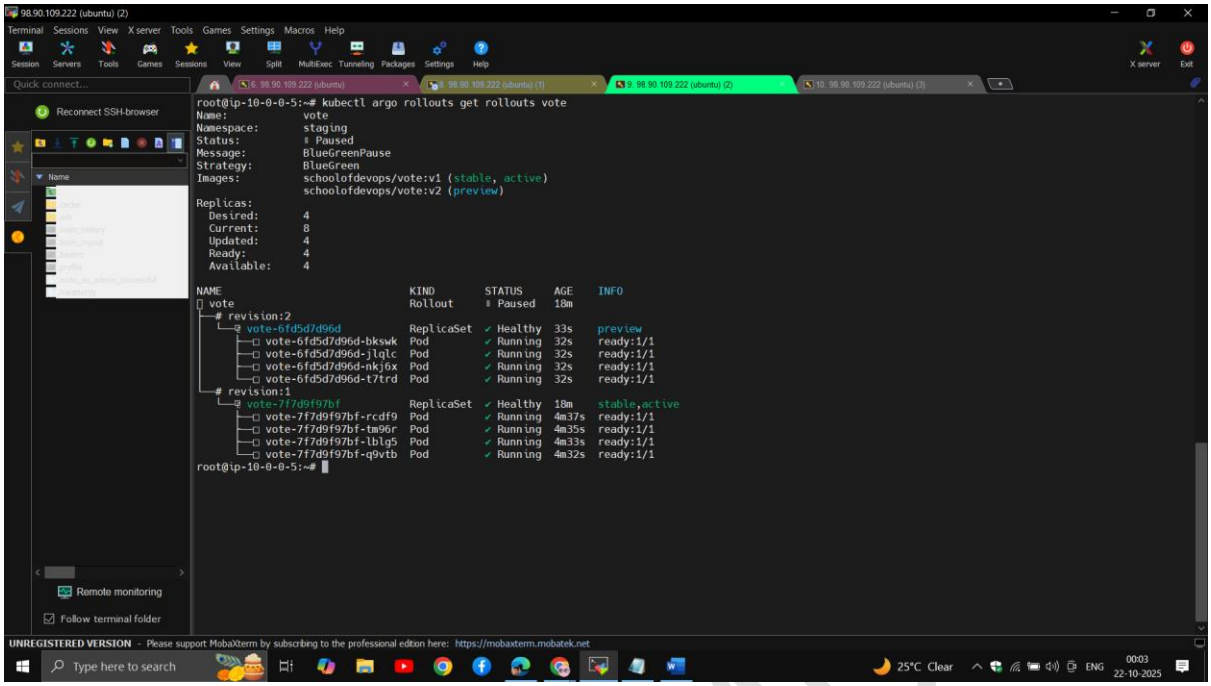

SecurityGroup | EC2 | us-east-1
Kubernetes Operational View 0.9
Nike vs Adidas!
Not secure
98.90.109.222:32000/#scale=3.0%252525252525252525252525252525252525252525252525252525252525...
Sign in
SORT: NAME
DEFAULT
https://10.96.0.1:443
kind-control-plan...
kind-worker
kind-worker2
Type here to search
25°C Clear
ENG
21:09
21-10-2025

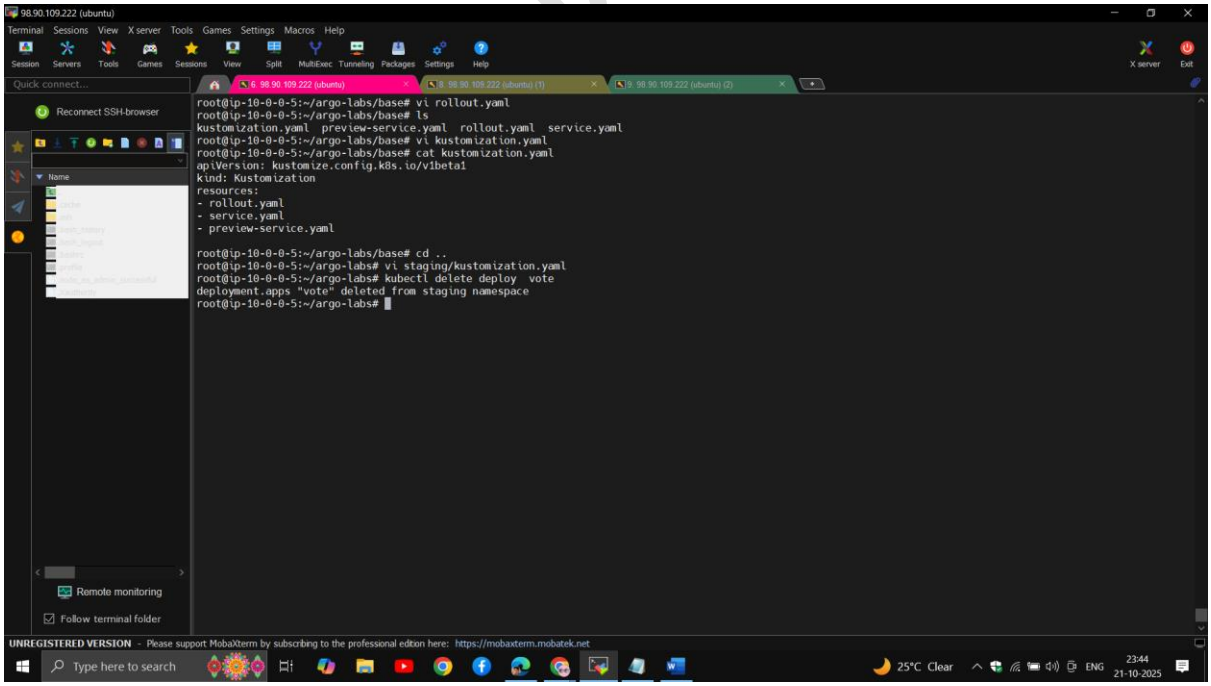

98.90.109.222 (ubuntu) (1)
Every 2.0s: kubectl get all,rollouts
ip-10-0-0-5: Tue Oct 21 18:12:46 2025
NAME READY STATUS RESTARTS AGE
pod/vote-697bcbdd7b-2122f 1/1 Running 0 163m
pod/vote-697bcbdd7b-gx5qt 1/1 Running 0 163m
NAME TYPE CLUSTER-IP EXTERNAL-IP PORT(S) AGE
service/vote NodePort 10.96.137.18 <none> 80:30000/TCP 163m
service/vote-preview NodePort 10.96.5.129 <none> 80:30100/TCP 116m
NAME READY UP-TO-DATE AVAILABLE AGE
deployment.apps/vote 2/2 2 2 163m
NAME DESIRED CURRENT READY AGE
replicaset.apps/vote-697bcbdd7b 2 2 2 163m
UNREGISTERED VERSION - Please support MobaXterm by subscribing to the professional edition here: https://mobaxterm.mobatek.net

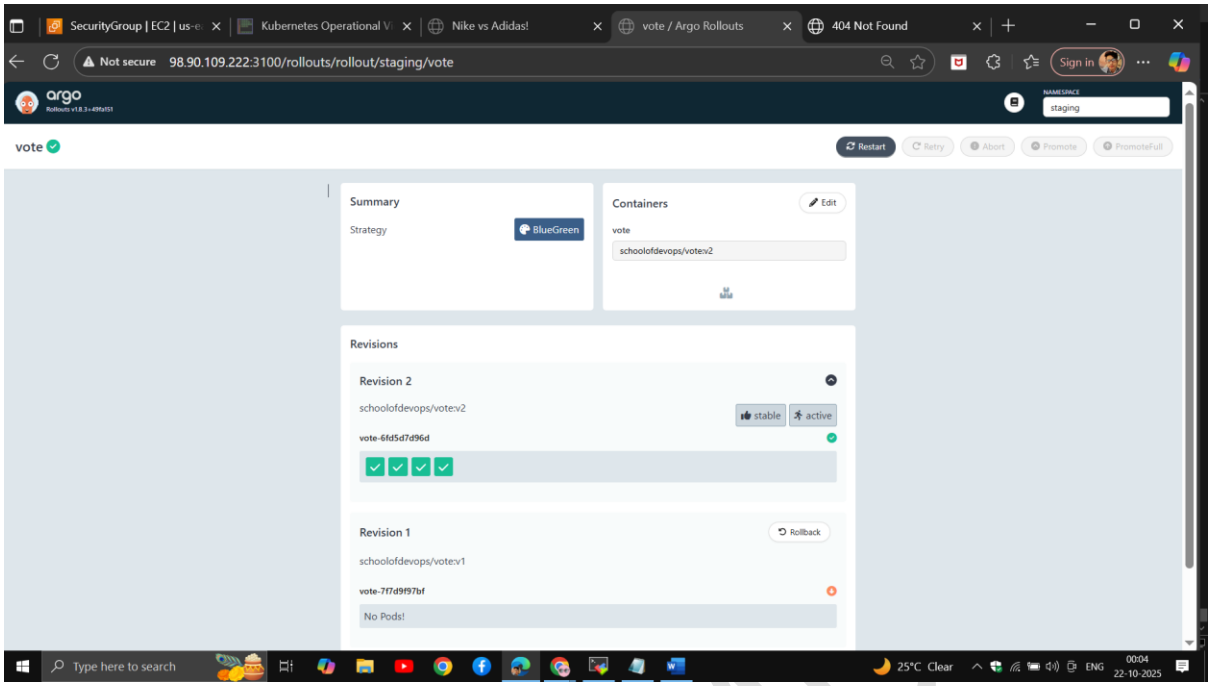

98.90.109.222 (ubuntu) (1)
Every 2.0s: kubectl get all,rollouts
ip-10-0-0-5: Tue Oct 21 18:13:48 2025
NAME TYPE CLUSTER-IP EXTERNAL-IP PORT(S) AGE
service/vote NodePort 10.96.137.18 <none> 80:30000/TCP 164m
service/vote-preview NodePort 10.96.5.129 <none> 80:30100/TCP 117m
Remote monitoring
Follow terminal folder
UNREGISTERED VERSION - Please support MobaXterm by subscribing to the professional edition here: https://mobaxterm.mobatek.net

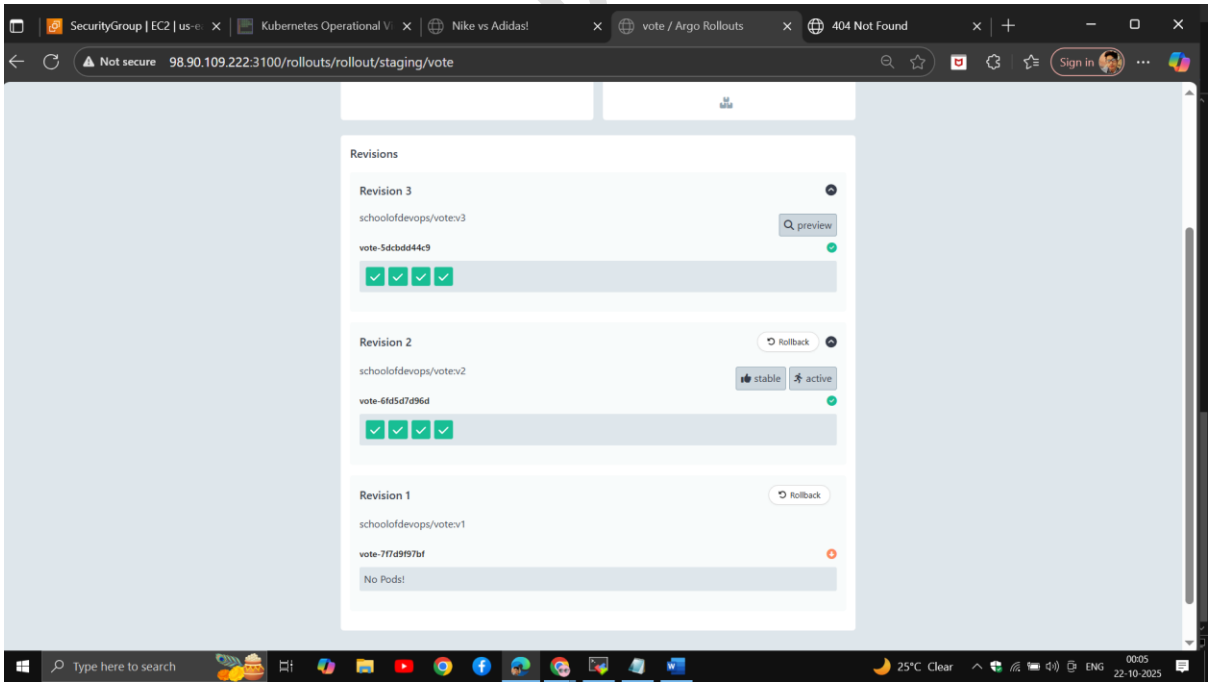

98.90.109.222 (ubuntu) (2)
root@ip-10-0-0-5:~# kubectl argo rollouts get rollouts vote
Name: vote
Namespace: staging
Status: ✔ Healthy
Strategy: BlueGreen
Images: schoolofdevops/vote:v1 (stable, active)
Replicas:
Desired: 4
Current: 4
Updated: 4
Ready: 4
Available: 4
NAME KIND STATUS AGE INFO
vote Rollout ✔ Healthy 3m19s
revision:1
vote-7f7d9f97bf ReplicaSet ✔ Healthy 3m19s stable,active
vote-7f7d9f97bf-57472 Pod ✔ Running 3m18s ready:1/1
vote-7f7d9f97bf-g8b62 Pod ✔ Running 3m18s ready:1/1
vote-7f7d9f97bf-hfk88 Pod ✔ Running 3m18s ready:1/1
vote-7f7d9f97bf-r7b8j Pod ✔ Running 3m18s ready:1/1
root@ip-10-0-0-5:~#
UNREGISTERED VERSION - Please support MobaXterm by subscribing to the professional edition here: https://mobaxterm.mobatek.net

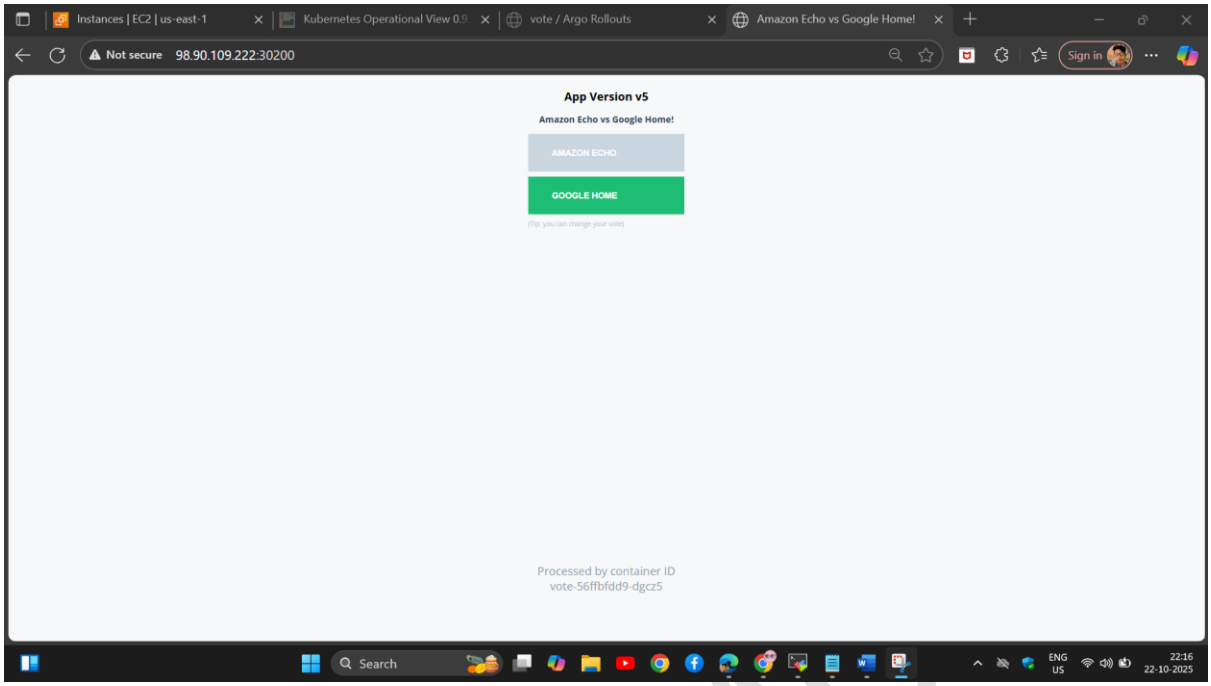

98.90.109.222 (ubuntu) (1)
Every 2.0s: kubectl get all,rollouts
ip-10-0-0-5: Tue Oct 21 18:15:11 2025
NAME READY STATUS RESTARTS AGE
pod/vote-7f7d9f97bf-57472 1/1 Running 0 21s
pod/vote-7f7d9f97bf-g8b62 1/1 Running 0 21s
pod/vote-7f7d9f97bf-hfk88 1/1 Running 0 21s
pod/vote-7f7d9f97bf-r7b8j 1/1 Running 0 21s
NAME TYPE CLUSTER-IP EXTERNAL-IP PORT(S) AGE
service/vote NodePort 10.96.137.18 <none> 80:30000/TCP 165m
service/vote-preview NodePort 10.96.5.129 <none> 80:30100/TCP 118m
NAME DESIRED CURRENT READY AGE
replicaset.apps/vote-7f7d9f97bf 4 4 4 22s
NAME DESIRED CURRENT UP-TO-DATE AVAILABLE AGE
rollout.argoproj.io/vote 4 4 4 4 22s
Remote monitoring
Follow terminal folder
UNREGISTERED VERSION - Please support MobaXterm by subscribing to the professional edition here: https://mobaxterm.mobatek.net

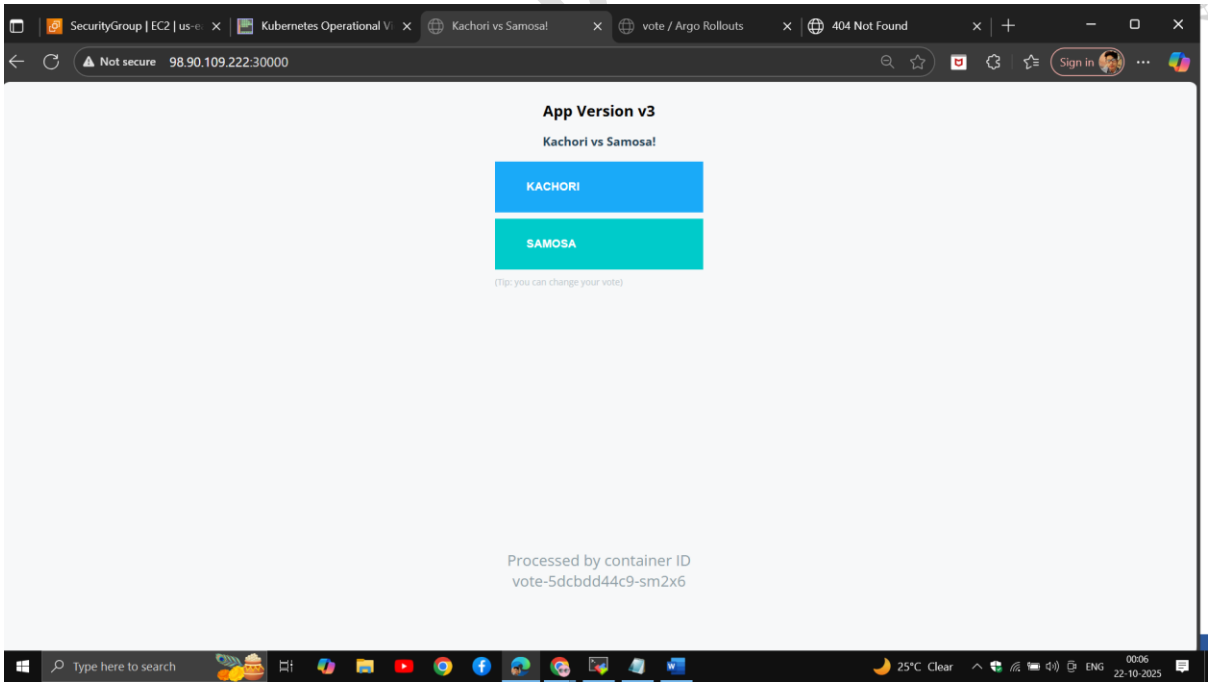

98.90.109.222 (ubuntu) (2)
root@ip-10-0-0-5:~# kubectl argo rollouts get rollouts vote
Name: vote
Namespace: staging
Status: ॥ Paused
Message: BlueGreenPause
Strategy: BlueGreen
Images: schoolofdevops/vote:v1 (stable, active)
schoolofdevops/vote:v2 (preview)
Replicas:
Desired: 4
Current: 8
Updated: 4
Ready: 4
Available: 4
NAME KIND STATUS AGE INFO
vote Rollout ॥ Paused 18m
revision:2
vote-6fd5d7d96d ReplicaSet ✓ Healthy 33s preview
vote-6fd5d7d96d-bkswk Pod ✓ Running 32s ready:1/1
vote-6fd5d7d96d-jlqlc Pod ✓ Running 32s ready:1/1
vote-6fd5d7d96d-nkj6x Pod ✓ Running 32s ready:1/1
vote-6fd5d7d96d-t7trd Pod ✓ Running 32s ready:1/1
revision:1
vote-7f7d9f97bf ReplicaSet ✓ Healthy 18m stable,active
vote-7f7d9f97bf-rcdf9 Pod ✓ Running 4m37s ready:1/1
vote-7f7d9f97bf-tm96r Pod ✓ Running 4m35s ready:1/1
vote-7f7d9f97bf-lblg5 Pod ✓ Running 4m33s ready:1/1
vote-7f7d9f97bf-q9vtb Pod ✓ Running 4m32s ready:1/1
root@ip-10-0-0-5:~#
Reconnect SSH-browser
Remote monitoring
Follow terminal folder
UNREGISTERED VERSION - Please support MobaXterm by subscribing to the professional edition here: https://mobaxterm.mobatek.net
Type here to search
25°C Clear
ENG
00:03
22-10-2025

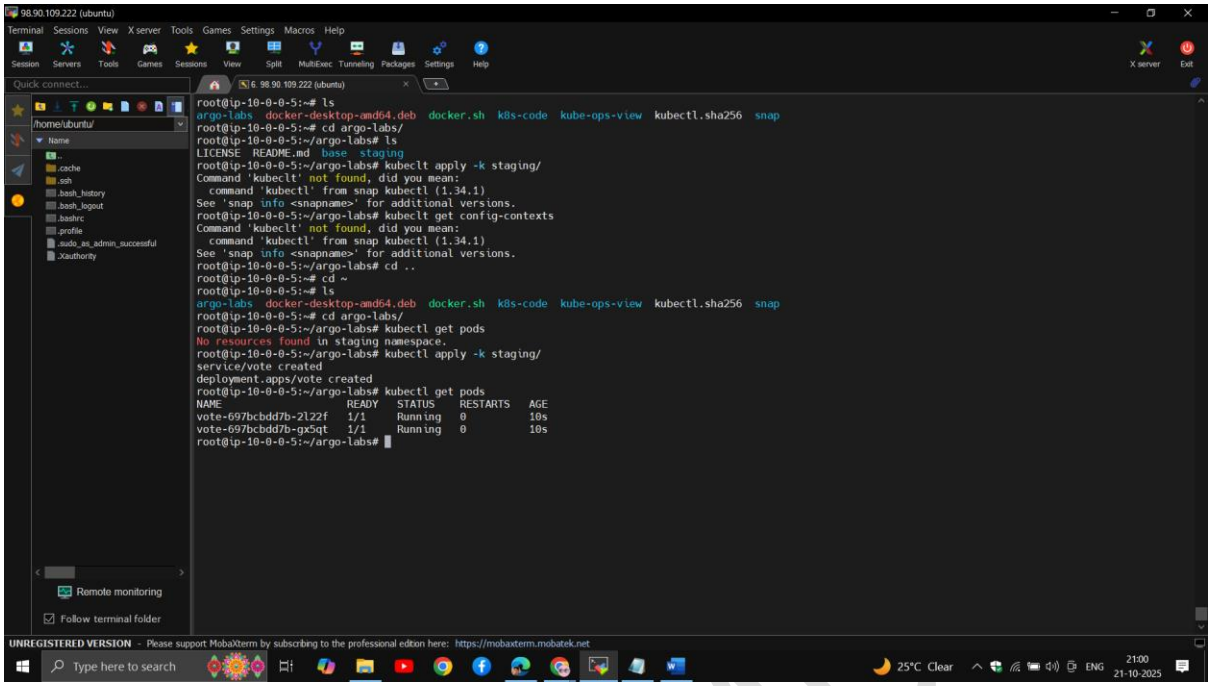

98.90.109.222 (ubuntu)
root@ip-10-0-0-5:~/argo-labs/base# vi rollout.yaml
root@ip-10-0-0-5:~/argo-labs/base# ls
kustomization.yaml preview-service.yaml rollout.yaml service.yaml
root@ip-10-0-0-5:~/argo-labs/base# vi kustomization.yaml
root@ip-10-0-0-5:~/argo-labs/base# cat kustomization.yaml
apiVersion: kustomize.config.k8s.io/v1beta1
kind: Kustomization
resources:
- rollout.yaml
- service.yaml
- preview-service.yaml
root@ip-10-0-0-5:~/argo-labs/base# cd ..
root@ip-10-0-0-5:~/argo-labs# vi staging/kustomization.yaml
root@ip-10-0-0-5:~/argo-labs# kubectl delete deploy vote
deployment.apps "vote" deleted from staging namespace
root@ip-10-0-0-5:~/argo-labs#
Reconnect SSH-browser
Remote monitoring
Follow terminal folder
UNREGISTERED VERSION - Please support MobaXterm by subscribing to the professional edition here: https://mobaxterm.mobatek.net
Type here to search
25°C Clear
ENG
23:44
21-10-2025

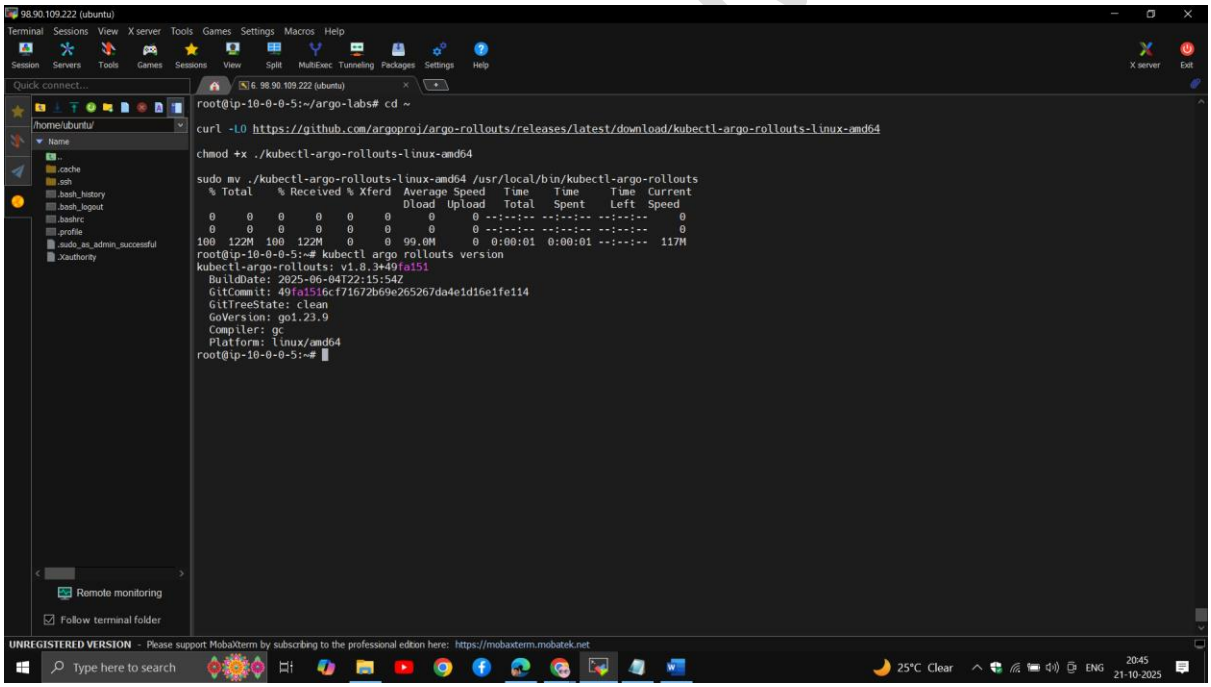

SecurityGroup | EC2 | us-e
Kubernetes Operational Vi
Nike vs Adidas!
vote / Argo Rollouts
404 Not Found
Not secure
98.90.109.222:3100/rollouts/rollout/staging/vote
Sign in
argo
NAMESPACE
staging
vote
Restart
Retry
Abort
Promote
PromoteFull
Summary
Strategy
BlueGreen
Containers
Edit
vote
schoolofdevops/vote:v2
Revisions
Revision 2
schoolofdevops/vote:v2
stable
active
vote-6fd5d7d96d
Revision 1
Rollback
schoolofdevops/vote:v1
vote-7f7d9f97bf
No Pods!
Type here to search
25°C Clear
ENG
00:04
22-10-2025

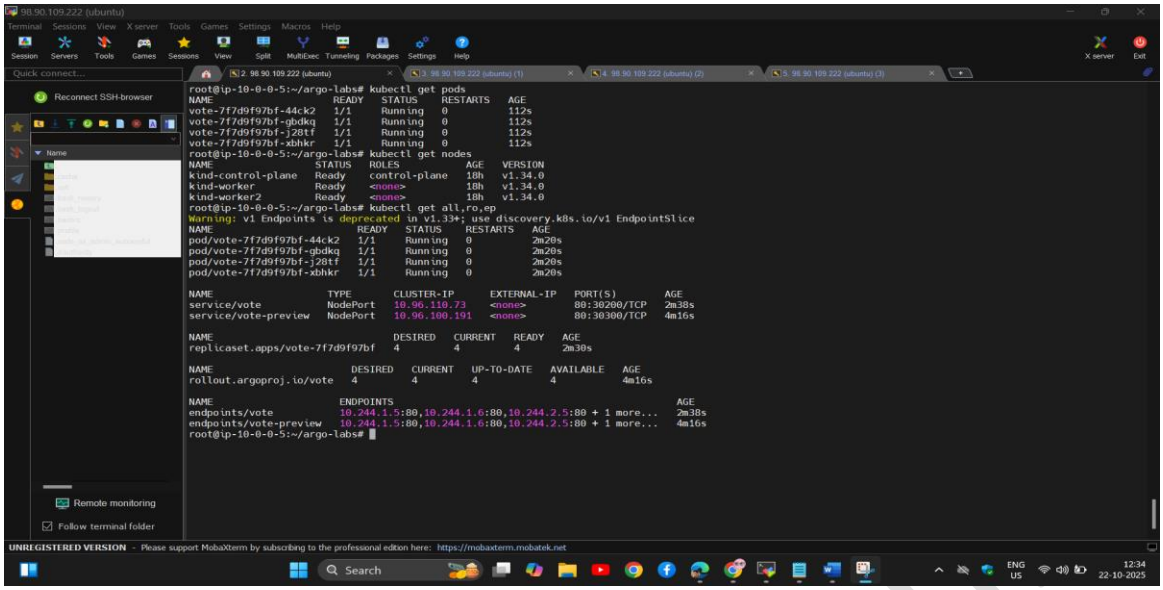

SecurityGroup | EC2 | us-e
Kubernetes Operational Vi
Nike vs Adidas!
vote / Argo Rollouts
404 Not Found
Not secure
98.90.109.222:3100/rollouts/rollout/staging/vote
Sign in
Revisions
Revision 3
schoolofdevops/vote:v3
preview
vote-5dcbdd44c9
Revision 2
Rollback
schoolofdevops/vote:v2
stable
active
vote-6fd5d7d96d
Revision 1
Rollback
schoolofdevops/vote:v1
vote-7f7d9f97bf
No Pods!
Type here to search
25°C Clear
ENG
00:05
22-10-2025

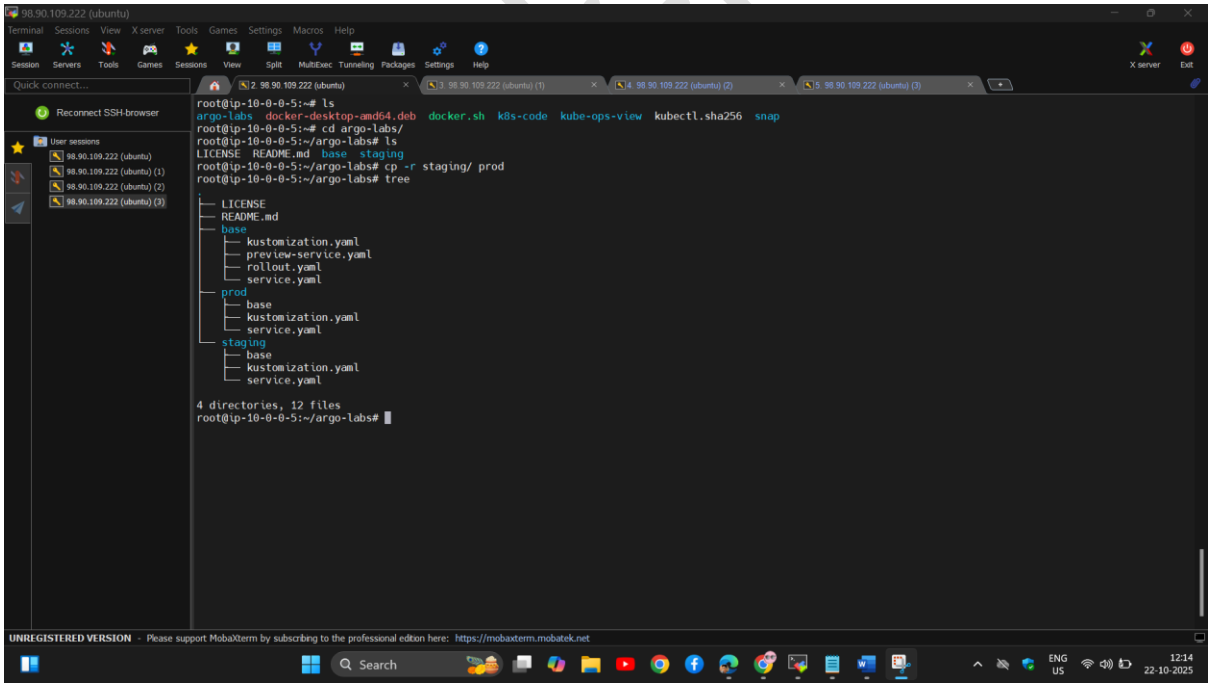

Instances | EC2 | us-east-1
Kubernetes Operational View 0.9
vote / Argo Rollouts
Amazon Echo vs Google Home!
Not secure 98.90.109.222:30200
Sign in
App Version v5
Amazon Echo vs Google Home!
AMAZON ECHO
GOOGLE HOME
(Tip: you can change your vote)
Processed by container ID
vote-56ffbfdd9-dgcz5
Search
ENG
US
22:16
22-10-2025

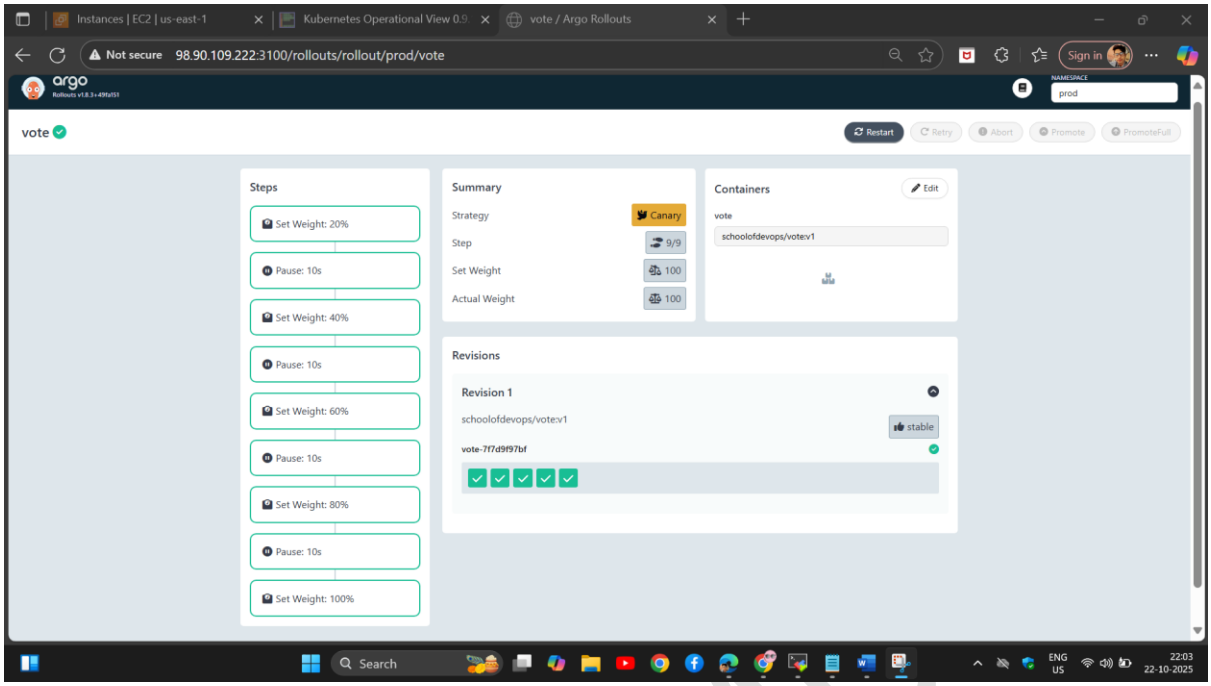

SecurityGroup | EC2 | us-e
Kubernetes Operational Vi
Kachori vs Samosa!
vote / Argo Rollouts
404 Not Found
Not secure 98.90.109.222:30000
Sign in
App Version v3
Kachori vs Samosa!
KACHORI
SAMOSA
(Tip: you can change your vote)
Processed by container ID
vote-5dcbdd44c9-sm2x6
Type here to search
25°C Clear
ENG
00:06
22-10-2025

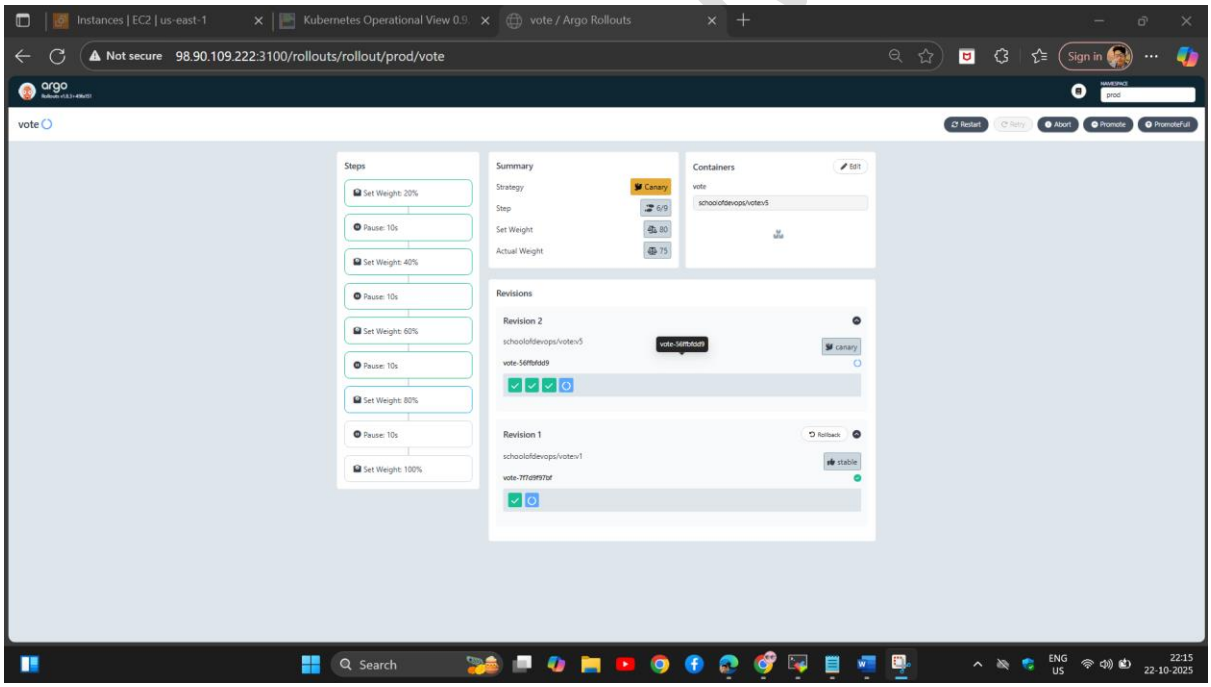

98.90.109.222 (ubuntu)
root@ip-10-0-0-5:~# ls
argo-labs docker-desktop-amd64.deb docker.sh k8s-code kube-ops-view kubectl.sha256 snap
root@ip-10-0-0-5:~# cd argo-labs/
root@ip-10-0-0-5:~/argo-labs# ls
LICENSE README.md base staging
root@ip-10-0-0-5:~/argo-labs# kubeclt apply -k staging/
Command 'kubeclt' not found, did you mean:
command 'kubectl' from snap kubectl (1.34.1)
See 'snap info <snapname>' for additional versions.
root@ip-10-0-0-5:~/argo-labs# kubeclt get config-contexts
Command 'kubeclt' not found, did you mean:
command 'kubectl' from snap kubectl (1.34.1)
See 'snap info <snapname>' for additional versions.
root@ip-10-0-0-5:~/argo-labs# cd ..
root@ip-10-0-0-5:~# cd ~
root@ip-10-0-0-5:~# ls
argo-labs docker-desktop-amd64.deb docker.sh k8s-code kube-ops-view kubectl.sha256 snap
root@ip-10-0-0-5:~# cd argo-labs/
root@ip-10-0-0-5:~/argo-labs# kubectl get pods
No resources found in staging namespace.
root@ip-10-0-0-5:~/argo-labs# kubectl apply -k staging/
service/vote created
deployment.apps/vote created
root@ip-10-0-0-5:~/argo-labs# kubectl get pods
NAME READY STATUS RESTARTS AGE
vote-697bcbdd7b-2l22f 1/1 Running 0 10s
vote-697bcbdd7b-gx5qt 1/1 Running 0 10s
root@ip-10-0-0-5:~/argo-labs#
Remote monitoring
Follow terminal folder
UNREGISTERED VERSION - Please support MobaXterm by subscribing to the professional edition here: https://mobaxterm.mobatek.net
Type here to search
25°C Clear
ENG
21:00
21-10-2025

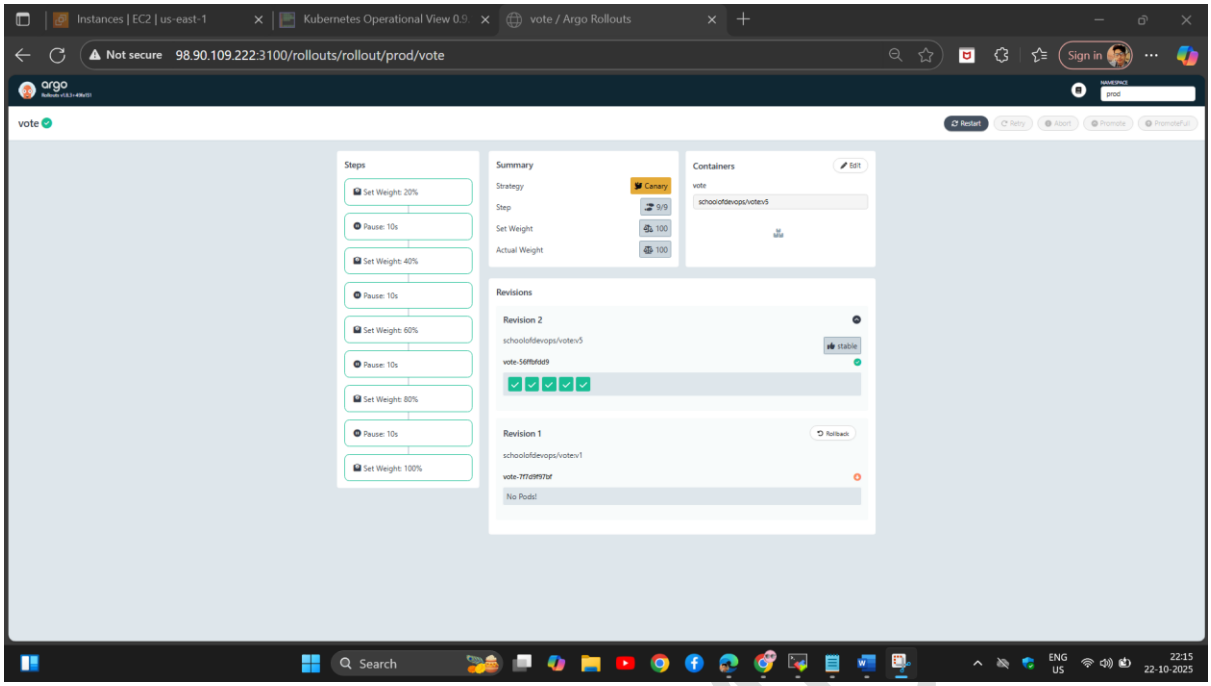

98.90.109.222 (ubuntu)
root@ip-10-0-0-5:~/argo-labs# cd ~
curl -LO https://github.com/argoproj/argo-rollouts/releases/latest/download/kubectl-argo-rollouts-linux-amd64
chmod +x ./kubectl-argo-rollouts-linux-amd64
sudo mv ./kubectl-argo-rollouts-linux-amd64 /usr/local/bin/kubectl-argo-rollouts
% Total % Received % Xferd Average Speed Time Time Time Current
Dload Upload Total Spent Left Speed
0 0 0 0 0 0 0 0 --:--:-- --:--:-- --:--:-- 0
0 0 0 0 0 0 0 0 --:--:-- --:--:-- --:--:-- 0
100 122M 100 122M 0 0 99.0M 0 0:00:01 0:00:01 --:--:-- 117M
root@ip-10-0-0-5:~# kubectl argo rollouts version
kubectl-argo-rollouts: v1.8.3+49fa151
BuildDate: 2025-06-04T22:15:54Z
GitCommit: 49fa151fcf71672b69e265267da4e1d16e1fe114
GitTreeState: clean
GoVersion: go1.23.9
Compiler: gc
Platform: linux/amd64
root@ip-10-0-0-5:~#
Remote monitoring
Follow terminal folder
UNREGISTERED VERSION - Please support MobaXterm by subscribing to the professional edition here: https://mobaxterm.mobatek.net
Type here to search
25°C Clear
ENG
20:45
21-10-2025

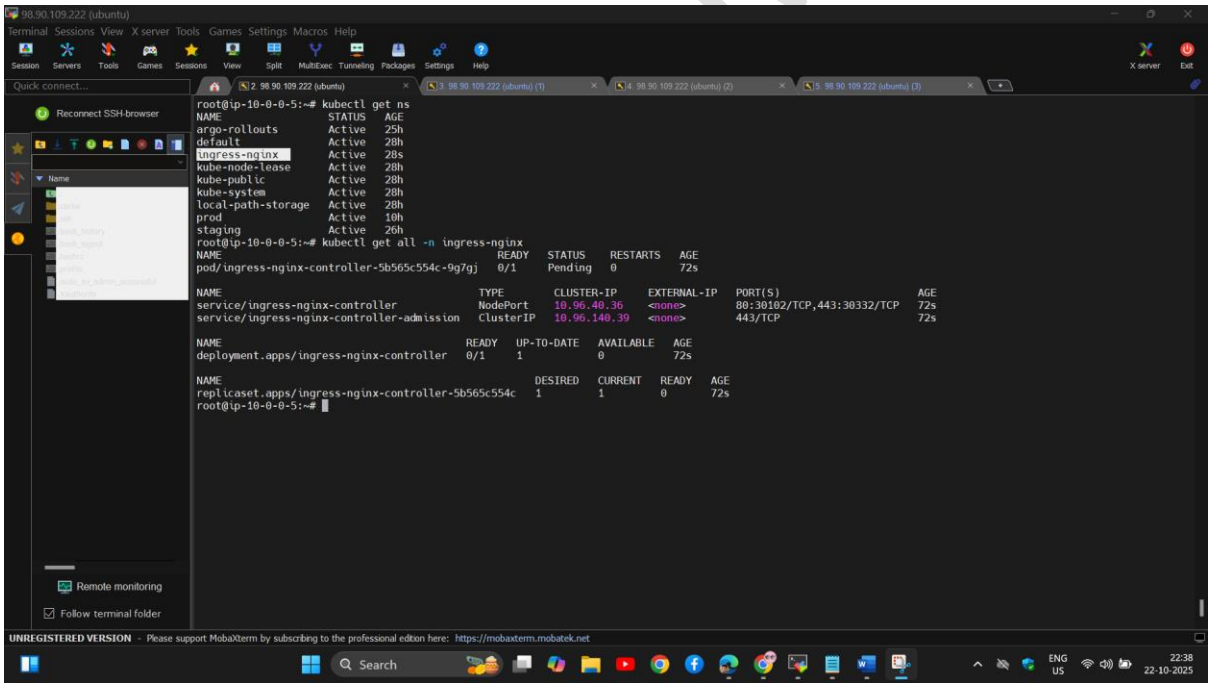

98.90.109.222 (ubuntu)
root@ip-10-0-0-5:~/argo-labs# kubectl get pods
NAME READY STATUS RESTARTS AGE
vote-7f7d9f97bf-44ck2 1/1 Running 0 112s
vote-7f7d9f97bf-gbdkq 1/1 Running 0 112s
vote-7f7d9f97bf-j28tf 1/1 Running 0 112s
vote-7f7d9f97bf-xbhkr 1/1 Running 0 112s
root@ip-10-0-0-5:~/argo-labs# kubectl get nodes
NAME STATUS ROLES AGE VERSION
kind-control-plane Ready control-plane 18h v1.34.0
kind-worker Ready <none> 18h v1.34.0
kind-worker2 Ready <none> 18h v1.34.0
root@ip-10-0-0-5:~/argo-labs# kubectl get all,ro,ep
Warning: v1 Endpoints is deprecated in v1.33+; use discovery.k8s.io/v1 EndpointSlice
NAME READY STATUS RESTARTS AGE
pod/vote-7f7d9f97bf-44ck2 1/1 Running 0 2m20s
pod/vote-7f7d9f97bf-gbdkq 1/1 Running 0 2m20s
pod/vote-7f7d9f97bf-j28tf 1/1 Running 0 2m20s
pod/vote-7f7d9f97bf-xbhkr 1/1 Running 0 2m20s
NAME TYPE CLUSTER-IP EXTERNAL-IP PORT(S) AGE
service/vote NodePort 10.96.110.73 <none> 80:30200/TCP 2m38s
service/vote-preview NodePort 10.96.100.191 <none> 80:30300/TCP 4m16s
NAME DESIRED CURRENT READY AGE
replicaset.apps/vote-7f7d9f97bf 4 4 4 2m30s
NAME DESIRED CURRENT UP-TO-DATE AVAILABLE AGE
rollout.argoproj.io/vote 4 4 4 4 4m16s
NAME ENDPOINTS AGE
endpoints/vote 10.244.1.5:80,10.244.1.6:80,10.244.2.5:80 + 1 more... 2m38s
endpoints/vote-preview 10.244.1.5:80,10.244.1.6:80,10.244.2.5:80 + 1 more... 4m16s
root@ip-10-0-0-5:~/argo-labs#
Reconnect SSH-browser
Remote monitoring
Follow terminal folder
UNREGISTERED VERSION - Please support MobaXterm by subscribing to the professional edition here: https://mobaxterm.mobatek.net
Search
ENG US
12:34 22-10-2025

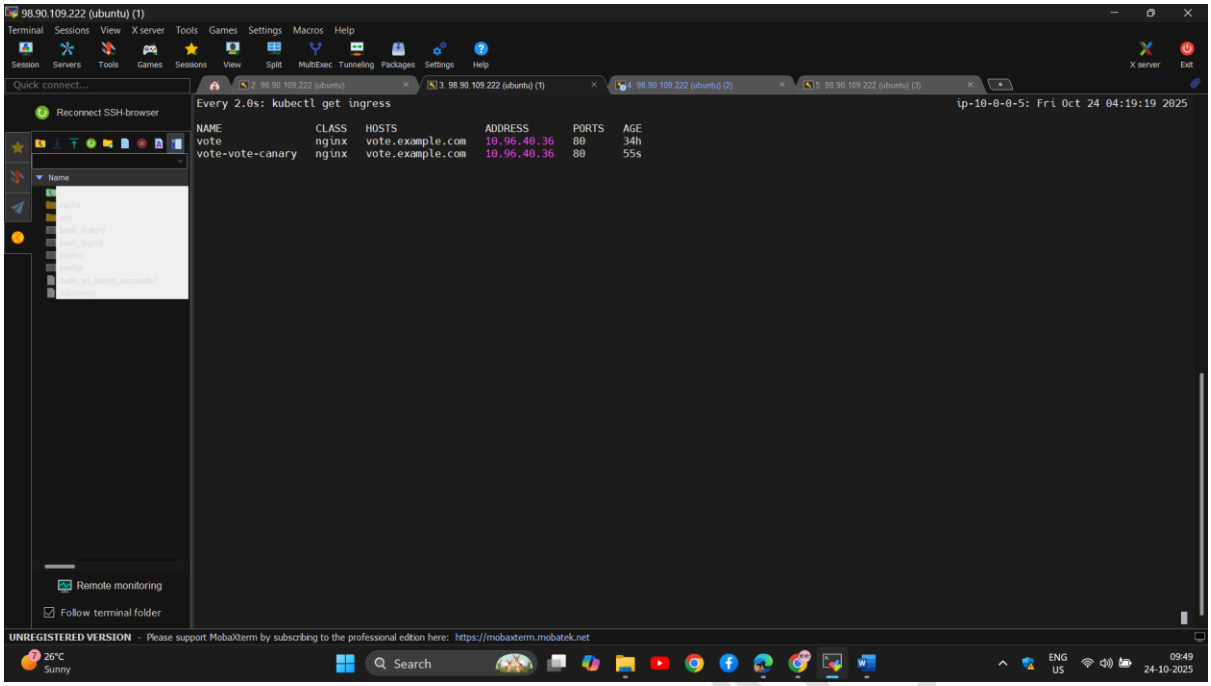

98.90.109.222 (ubuntu)
root@ip-10-0-0-5:~# ls
argo-labs docker-desktop-amd64.deb docker.sh k8s-code kube-ops-view kubectl.sha256 snap
root@ip-10-0-0-5:~# cd argo-labs/
root@ip-10-0-0-5:~/argo-labs# ls
LICENSE README.md base staging
root@ip-10-0-0-5:~/argo-labs# cp -r staging/ prod
root@ip-10-0-0-5:~/argo-labs# tree
.
├── LICENSE
├── README.md
├── base
│ ├── kustomization.yaml
│ ├── preview-service.yaml
│ ├── rollout.yaml
│ └── service.yaml
├── prod
│ ├── base
│ ├── kustomization.yaml
│ └── service.yaml
└── staging
├── base
├── kustomization.yaml
└── service.yaml
4 directories, 12 files
root@ip-10-0-0-5:~/argo-labs#
Reconnect SSH-browser
User sessions
98.90.109.222 (ubuntu)
98.90.109.222 (ubuntu) (1)
98.90.109.222 (ubuntu) (2)
98.90.109.222 (ubuntu) (3)
UNREGISTERED VERSION - Please support MobaXterm by subscribing to the professional edition here: https://mobaxterm.mobatek.net
Search
ENG US
12:14 22-10-2025

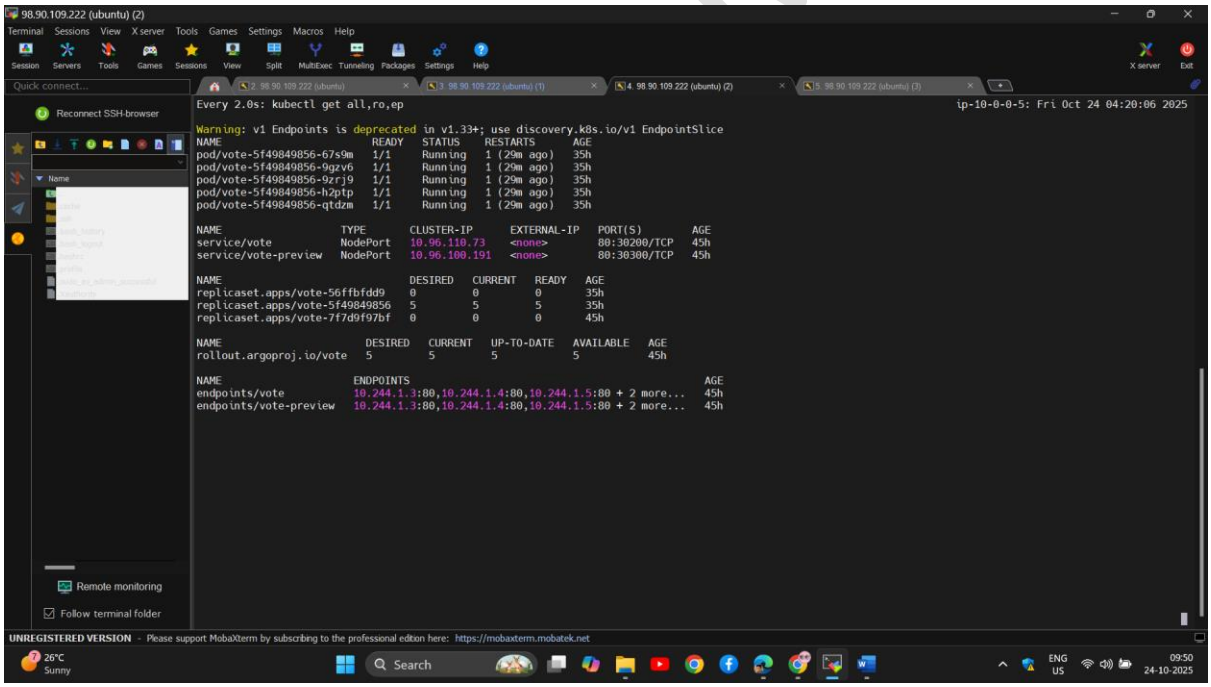

Instances | EC2 | us-east-1
Kubernetes Operational View 0.9
vote / Argo Rollouts
Not secure
98.90.109.222:3100/rollouts/rollout/prod/vote
Sign in
argo
NAMESPACE
prod
vote
Restart
Retry
Abort
Promote
PromoteFull
Steps
Set Weight: 20%
Pause: 10s
Set Weight: 40%
Pause: 10s
Set Weight: 60%
Pause: 10s
Set Weight: 80%
Pause: 10s
Set Weight: 100%
Summary
Strategy
Canary
Step
9/9
Set Weight
100
Actual Weight
100
Containers
Edit
vote
schoolofdevops/votev1
Revisions
Revision 1
schoolofdevops/votev1
stable
vote-7f7d9f97bf
Search
ENG
US
22:03
22-10-2025

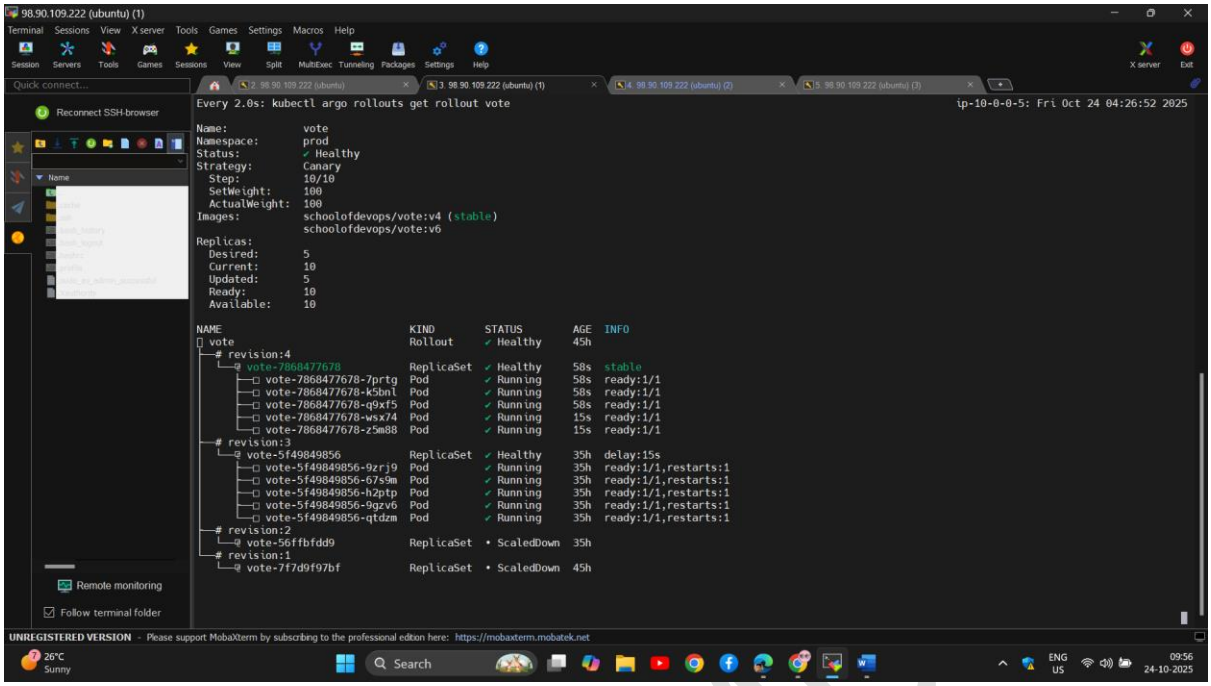

Instances | EC2 | us-east-1
Kubernetes Operational View 0.9
vote / Argo Rollouts
Not secure
98.90.109.222:3100/rollouts/rollout/prod/vote
Sign in
argo
prod
vote
Restart
Abort
Promote
PromoteFull
Steps
Set Weight: 20%
Pause: 10s
Set Weight: 40%
Pause: 10s
Set Weight: 60%
Pause: 10s
Set Weight: 80%
Pause: 10s
Set Weight: 100%
Summary
Strategy
Canary
Step
6/9
Set Weight
80
Actual Weight
75
Containers
Edit
vote
schoolofdevops/votev5
Revisions
Revision 2
schoolofdevops/votev5
canary
vote-56ffbfdd9
Revision 1
Rollback
schoolofdevops/votev1
stable
Search
ENG
US
22:15
22-10-2025

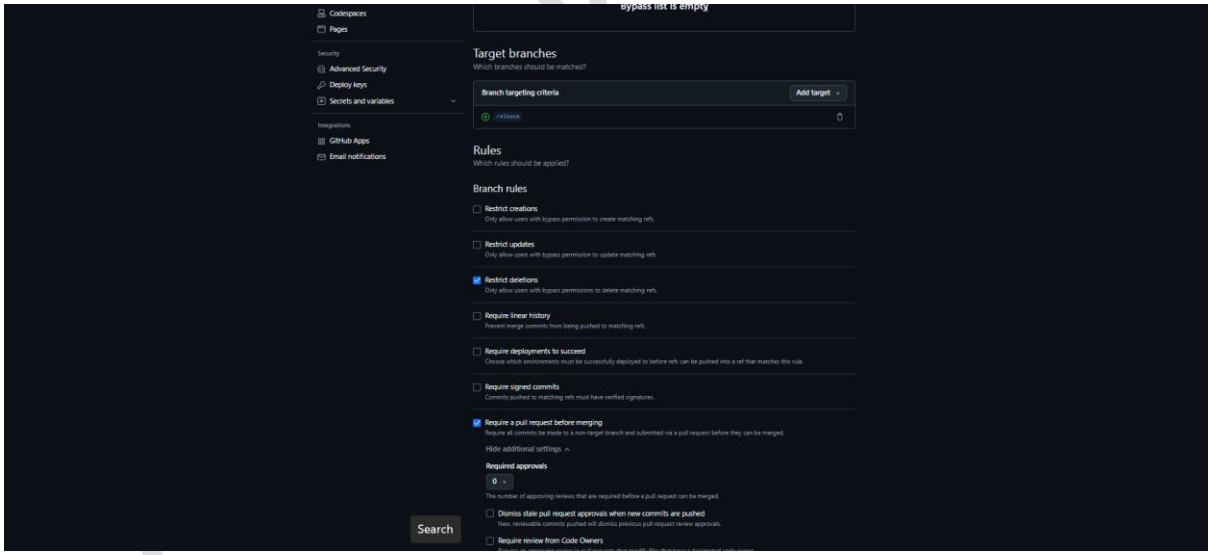

Instances | EC2 | us-east-1
Kubernetes Operational View 0.9
vote / Argo Rollouts
Not secure 98.90.109.222:3100/rollouts/rollout/prod/vote
Sign in
argo
vote
Restart
Steps
Set Weight: 20%
Pause: 10s
Set Weight: 40%
Pause: 10s
Set Weight: 60%
Pause: 10s
Set Weight: 80%
Pause: 10s
Set Weight: 100%
Summary
Strategy
Canary
Step
9/9
Set Weight
100
Actual Weight
100
Containers
Edit
vote
schoolofdevops/votev5
Revisions
Revision 2
schoolofdevops/votev5
stable
vote-56ffbfdd9
Revision 1
Rollback
schoolofdevops/votev1
vote-7f7d9f97bf
No Pods!
Search
ENG
US
22:15
22-10-2025

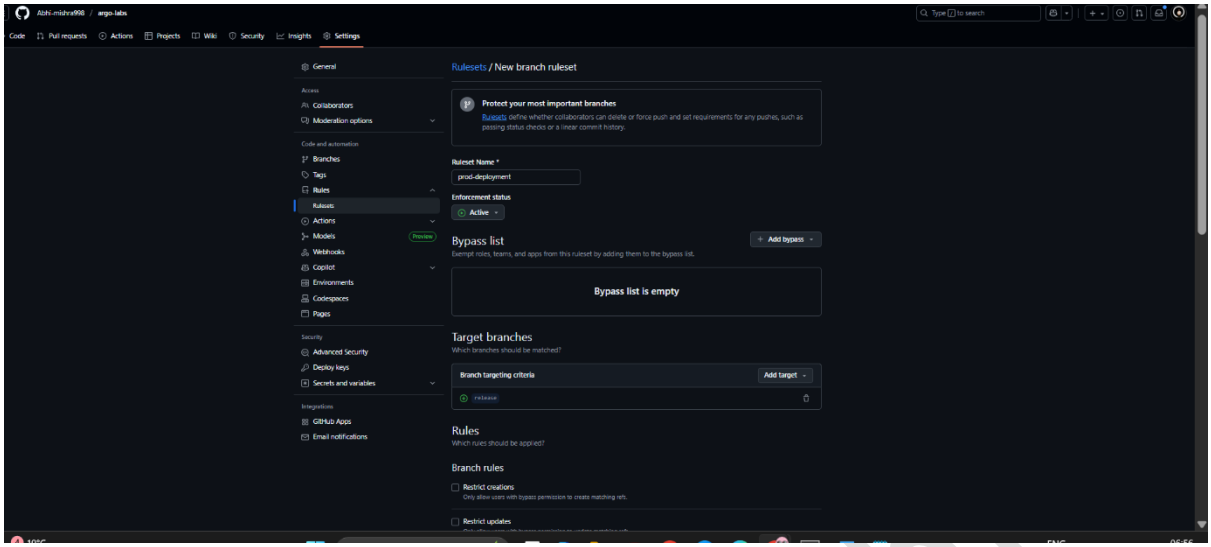

98.90.109.222 (ubuntu)
Terminal Sessions View X server Tools Games Settings Macros Help
Session Servers Tools Games Sessions View Split MultiExec Tunneling Packages Settings Help
X server Exit
Quick connect...
Reconnect SSH-browser
root@ip-10-0-0-5:~# kubectl get ns
NAME STATUS AGE
argo-rollouts Active 25h
default Active 28h
ingress-nginx Active 28s
kube-node-lease Active 28h
kube-public Active 28h
kube-system Active 28h
local-path-storage Active 28h
prod Active 10h
staging Active 26h
root@ip-10-0-0-5:~# kubectl get all -n ingress-nginx
NAME READY STATUS RESTARTS AGE
pod/ingress-nginx-controller-5b565c554c-9g7gj 0/1 Pending 0 72s
NAME TYPE CLUSTER-IP EXTERNAL-IP PORT(S) AGE
service/ingress-nginx-controller NodePort 10.96.40.36 <none> 80:30102/TCP,443:30332/TCP 72s
service/ingress-nginx-controller-admission ClusterIP 10.96.140.39 <none> 443/TCP 72s
NAME READY UP-TO-DATE AVAILABLE AGE
deployment.apps/ingress-nginx-controller 0/1 1 0 72s
NAME DESIRED CURRENT READY AGE
replicaset.apps/ingress-nginx-controller-5b565c554c 1 1 0 72s
root@ip-10-0-0-5:~#
Remote monitoring
Follow terminal folder
UNREGISTERED VERSION - Please support MobaXterm by subscribing to the professional edition here: https://mobaxterm.mobatek.net
Search
ENG
US
22:38
22-10-2025

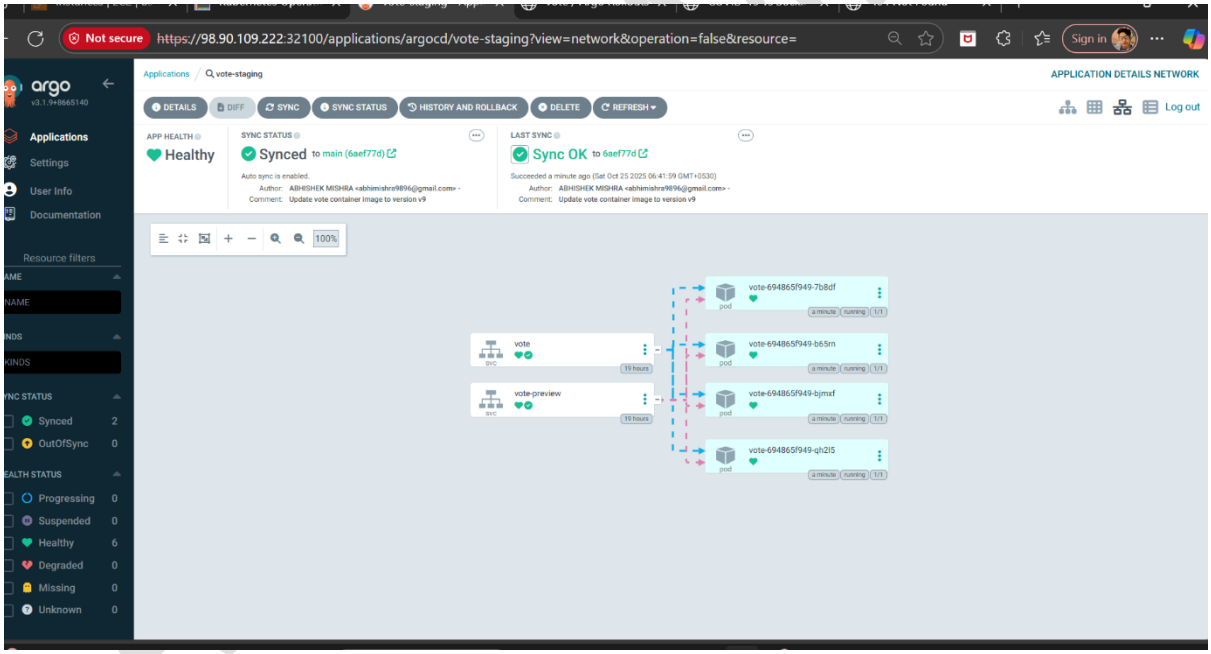

98.90.109.222 (ubuntu) (1)
Every 2.0s: kubectl get ingress
ip-10-0-0-5: Fri Oct 24 04:19:19 2025
NAME CLASS HOSTS ADDRESS PORTS AGE
vote nginx vote.example.com 10.96.40.36 80 34h
vote-vote-canary nginx vote.example.com 10.96.40.36 80 55s
Remote monitoring
Follow terminal folder
UNREGISTERED VERSION - Please support MobaXterm by subscribing to the professional edition here: https://mobaxterm.mobatek.net

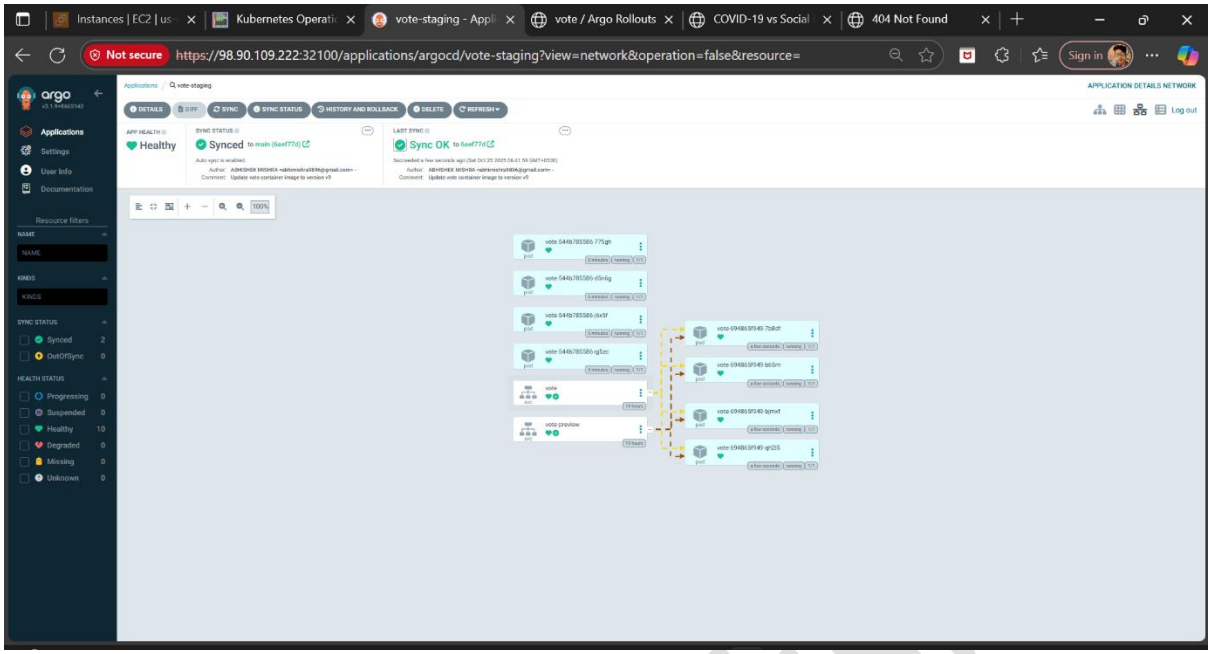

98.90.109.222 (ubuntu) (2)
Every 2.0s: kubectl get all,ro,ep
ip-10-0-0-5: Fri Oct 24 04:20:06 2025
Warning: v1 Endpoints is deprecated in v1.33+; use discovery.k8s.io/v1 EndpointSlice
pod/vote-5f49849856-67s9m 1/1 Running 1 (29m ago) 35h
pod/vote-5f49849856-9gzy6 1/1 Running 1 (29m ago) 35h
pod/vote-5f49849856-9zrj9 1/1 Running 1 (29m ago) 35h
pod/vote-5f49849856-h2ptp 1/1 Running 1 (29m ago) 35h
pod/vote-5f49849856-qtdzm 1/1 Running 1 (29m ago) 35h
service/vote NodePort 10.96.110.73 <none> 80:30200/TCP 45h
service/vote-preview NodePort 10.96.100.191 <none> 80:30300/TCP 45h
replicaset.apps/vote-56ffbfdd9 0 0 0 35h
replicaset.apps/vote-5f49849856 5 5 5 35h
replicaset.apps/vote-7f7d9f97bf 0 0 0 45h
rollout.argoproj.io/vote 5 5 5 5 45h
endpoints/vote 10.244.1.3:80,10.244.1.4:80,10.244.1.5:80 + 2 more... 45h
endpoints/vote-preview 10.244.1.3:80,10.244.1.4:80,10.244.1.5:80 + 2 more... 45h
Remote monitoring
Follow terminal folder
UNREGISTERED VERSION - Please support MobaXterm by subscribing to the professional edition here: https://mobaxterm.mobatek.net

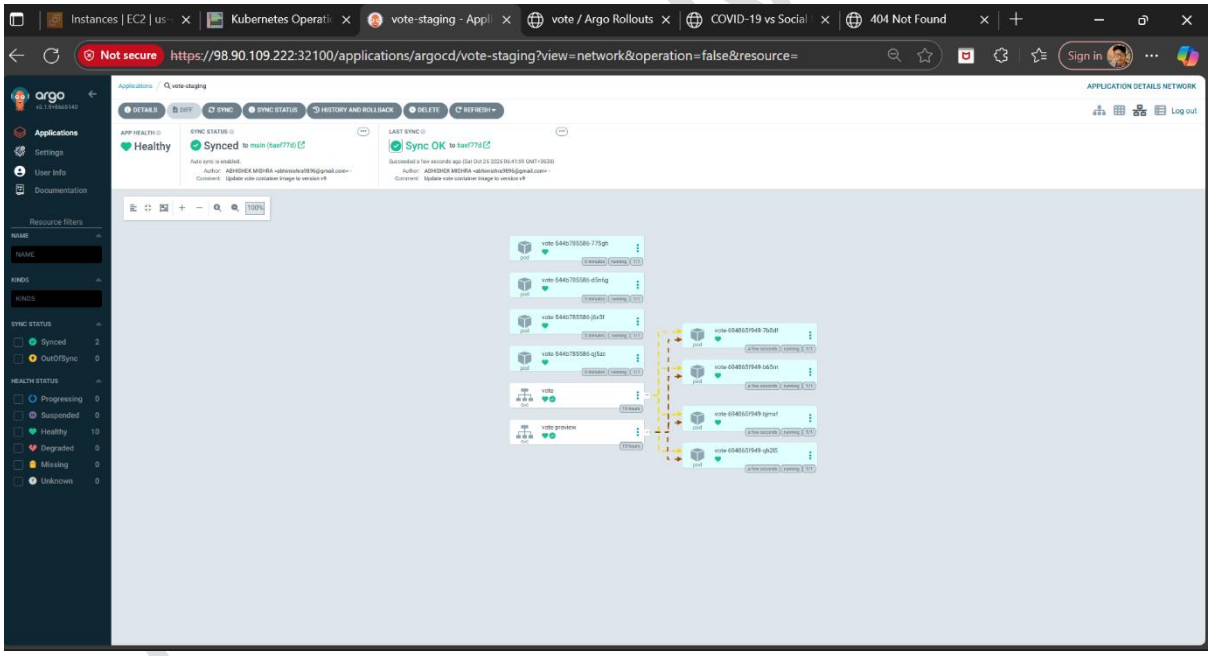

98.90.109.222 (ubuntu) (1)
Every 2.0s: kubectl argo rollouts get rollout vote
ip-10-0-0-5: Fri Oct 24 04:26:52 2025
Name: vote
Namespace: prod
Status: ✔ Healthy
Strategy: Canary
Step: 10/10
SetWeight: 100
ActualWeight: 100
Images: schoolofdevops/vote:v4 (stable)
schoolofdevops/vote:v6
Replicas:
Desired: 5
Current: 10
Updated: 5
Ready: 10
Available: 10
NAME KIND STATUS AGE INFO
vote Rollout ✔ Healthy 45h
revision:4
vote-7868477678 ReplicaSet ✔ Healthy 58s stable
vote-7868477678-7prtg Pod ✔ Running 58s ready:1/1
vote-7868477678-k5bnl Pod ✔ Running 58s ready:1/1
vote-7868477678-q9xf5 Pod ✔ Running 58s ready:1/1
vote-7868477678-wsx74 Pod ✔ Running 15s ready:1/1
vote-7868477678-z5m88 Pod ✔ Running 15s ready:1/1
revision:3
vote-5f49849856 ReplicaSet ✔ Healthy 35h delay:15s
vote-5f49849856-9zrj9 Pod ✔ Running 35h ready:1/1,restarts:1
vote-5f49849856-67s9m Pod ✔ Running 35h ready:1/1,restarts:1
vote-5f49849856-h2ptp Pod ✔ Running 35h ready:1/1,restarts:1
vote-5f49849856-9gzv6 Pod ✔ Running 35h ready:1/1,restarts:1
vote-5f49849856-qtdzm Pod ✔ Running 35h ready:1/1,restarts:1
revision:2
vote-56ffbfdd9 ReplicaSet • ScaledDown 35h
revision:1
vote-7f7d9f97bf ReplicaSet • ScaledDown 45h
Reconnect SSH-browser
Remote monitoring
Follow terminal folder
UNREGISTERED VERSION - Please support MobaXterm by subscribing to the professional edition here: https://mobaxterm.mobatek.net
Search
ENG
US
09:56
24-10-2025

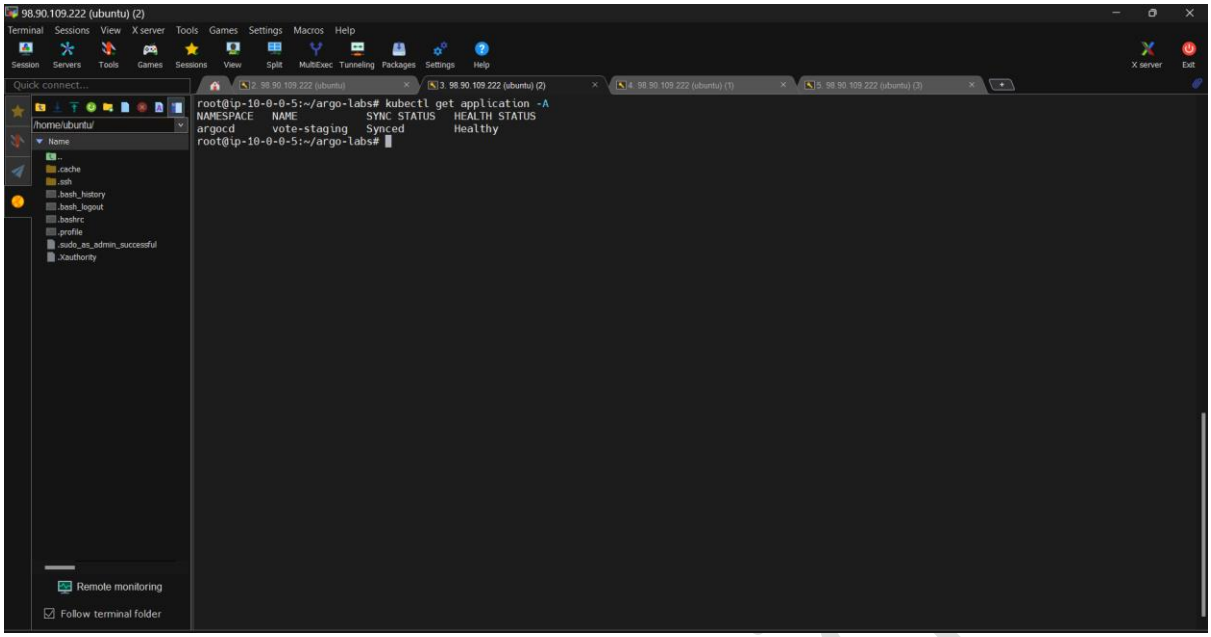

Codespaces
Pages
Advanced Security
Deploy keys
Secrets and variables
GitHub Apps
Email notifications
Target branches
Which branches should be matched?
Branch targeting criteria
Add target
release
Rules
Which rules should be applied?
Branch rules
Restrict creations
Only allow users with bypass permission to create matching refs.
Restrict updates
Only allow users with bypass permission to update matching refs.
Restrict deletions
Only allow users with bypass permissions to delete matching refs.
Require linear history
Prevent merge commits from being pushed to matching refs.
Require deployments to succeed
Require signed commits
Commits pushed to matching refs must have verified signatures.
Require a pull request before merging
Require all commits be made to a non-target branch and submitted via a pull request before they can be merged.
Hide additional settings
Required approvals
0
The number of approving reviews that are required before a pull request can be merged.
Dismiss stale pull request approvals when new commits are pushed
Require review from Code Owners
Search

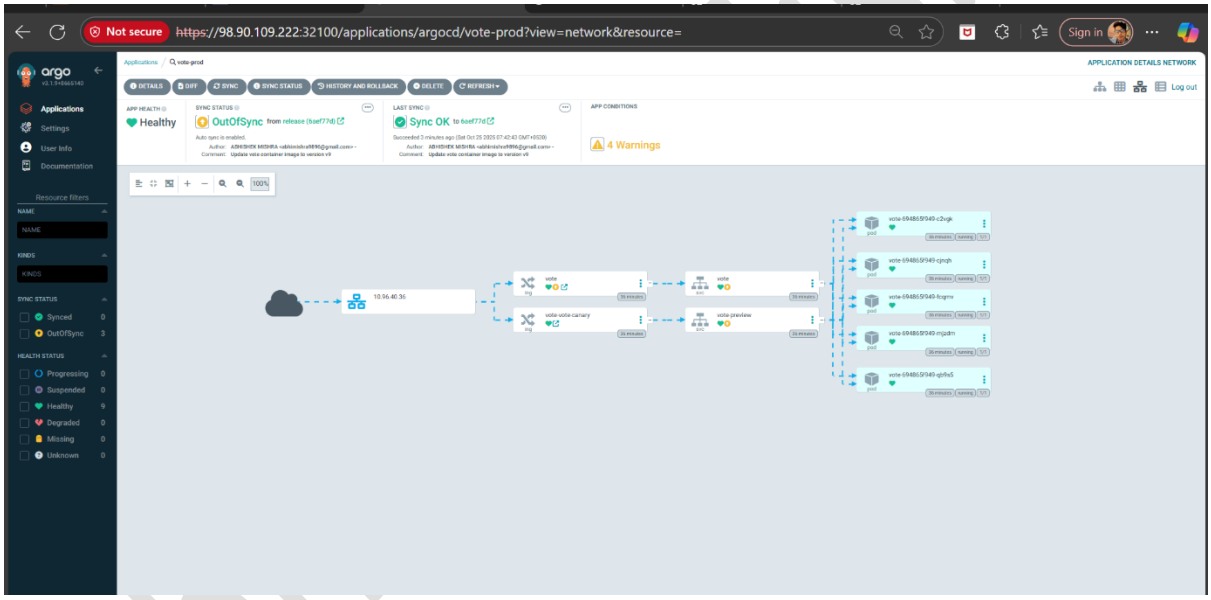

Abhi-mishra998 / argo-labs
Code
Pull requests
Actions
Projects
Wiki
Security
Insights
Settings
General
Access
Collaborators
Moderation options
Code and automation
Branches
Tags
Rules
Rulesets
Actions
Models
Preview
Webhooks
Copilot
Environments
Codespaces
Pages
Security
Advanced Security
Deploy keys
Secrets and variables
Integrations
GitHub Apps
Email notifications
Rulesets / New branch ruleset
Protect your most important branches
Rulesets define whether collaborators can delete or force push and set requirements for any pushes, such as passing status checks or a linear commit history.
Ruleset Name *
prod-deployment
Enforcement status
Active
Bypass list
Exempt roles, teams, and apps from this ruleset by adding them to the bypass list.
Add bypass
Bypass list is empty
Target branches
Which branches should be matched?
Branch targeting criteria
Add target
release
Rules
Which rules should be applied?
Branch rules
Restrict creations
Only allow users with bypass permission to create matching refs.
Restrict updates

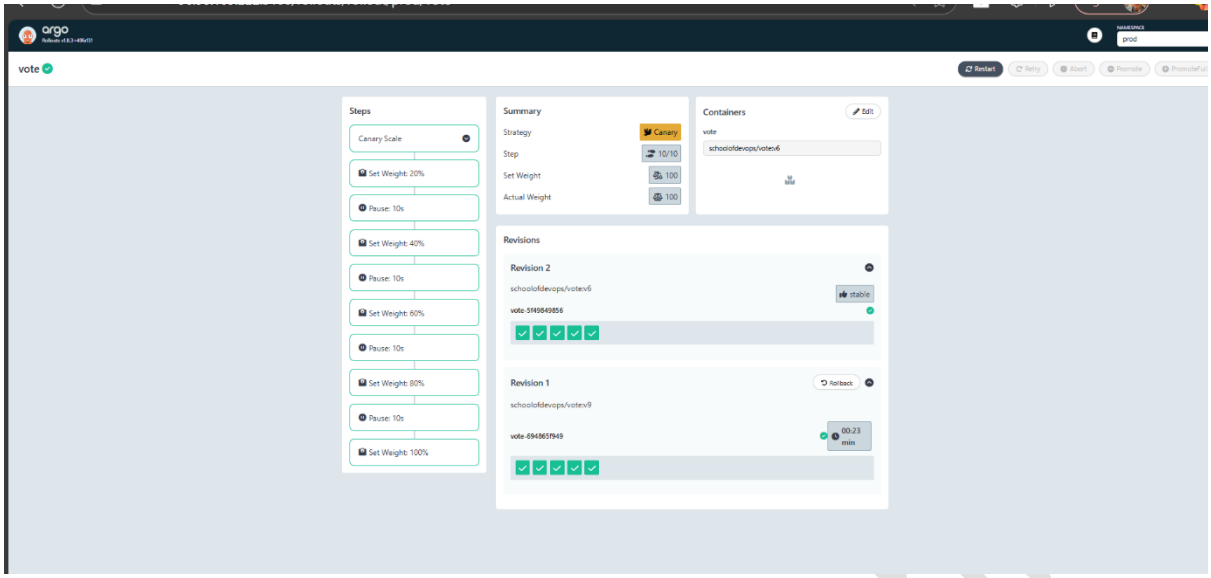

Not secure
https://98.90.109.222:32100/applications/argocd/vote-staging?view=network&operation=false&resource=
Sign in
argo
v3.1.9+8665140
Applications
Settings
User Info
Documentation
Resource filters
Synced 2
OutOfSync 0
Progressing 0
Suspended 0
Healthy 6
Degraded 0
Missing 0
Unknown 0
Applications
vote-staging
APPLICATION DETAILS NETWORK
DETAILS
DIFF
SYNC
SYNC STATUS
HISTORY AND ROLLBACK
DELETE
REFRESH
Log out
APP HEALTH
Healthy
SYNC STATUS
Synced to main (6aef77d)
Auto sync is enabled.
Author: ABHISHEK MISHRA <abhimishra9896@gmail.com> -
Comment: Update vote container image to version v9
LAST SYNC
Sync OK to 6aef77d
Succeeded a minute ago (Sat Oct 25 2025 06:41:59 GMT+0530)
Author: ABHISHEK MISHRA <abhimishra9896@gmail.com> -
Comment: Update vote container image to version v9
100%
vote
svc
19 hours
vote-preview
svc
19 hours
vote-694865f949-7b8df
vote-694865f949-b65rn
vote-694865f949-bjmxf
vote-694865f949-qh2l5
pod
a minute
running
1/1

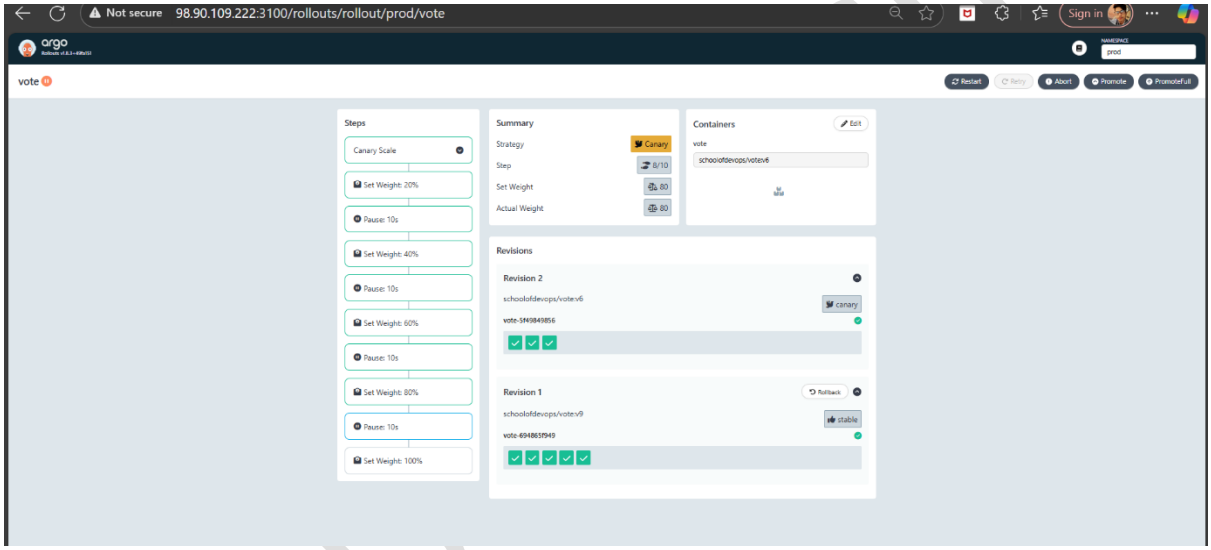

Not secure
https://98.90.109.222:32100/applications/argocd/vote-staging?view=network&operation=false&resource=
Sign in
argo
Applications
Settings
User Info
Documentation
Resource filters
APPLICATION DETAILS NETWORK
DETAILS
DIFF
SYNC
SYNC STATUS
HISTORY AND ROLLBACK
DELETE
REFRESH
Log out
APP HEALTH
Healthy
SYNC STATUS
Synced
LAST SYNC
Sync OK
Synced
OutOfSync
Progressing
Suspended
Healthy
Degraded
Missing
Unknown

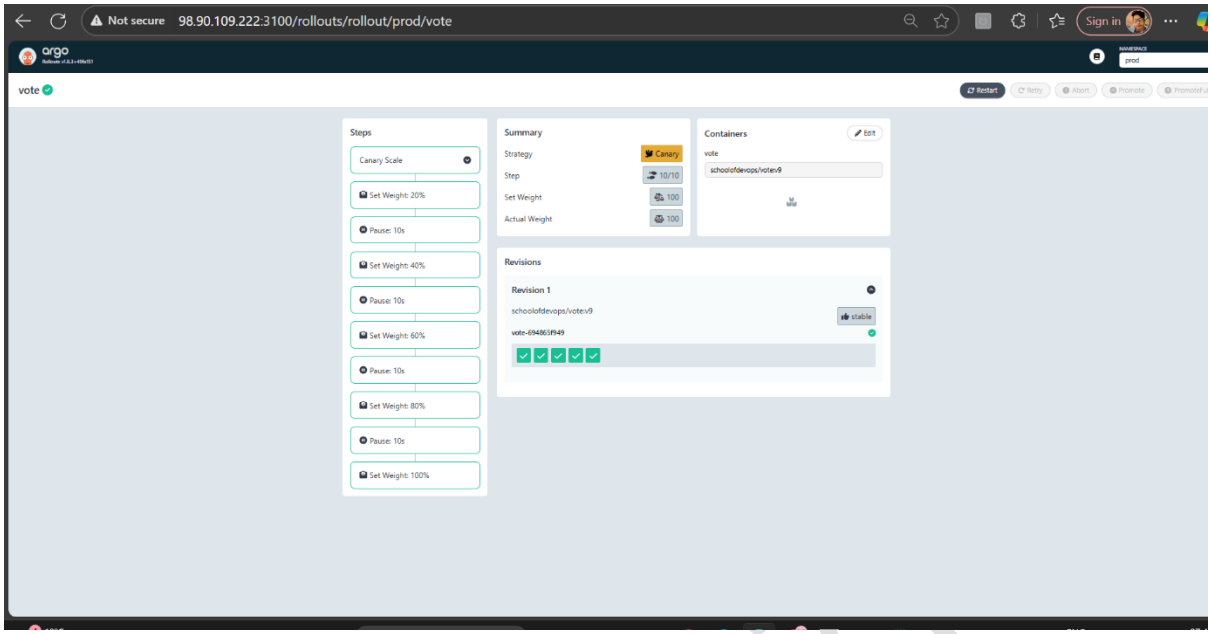

Not secure
https://98.90.109.222:32100/applications/argocd/vote-staging?view=network&operation=false&resource=
Sign in
argo
Applications
Settings
User Info
Documentation
Resource filters
APPLICATION DETAILS NETWORK
DETAILS
DIFF
SYNC
SYNC STATUS
HISTORY AND ROLLBACK
DELETE
REFRESH
Log out
APP HEALTH
Healthy
SYNC STATUS
Synced
LAST SYNC
Sync OK
Synced
OutOfSync
Progressing
Suspended
Healthy
Degraded
Missing
Unknown

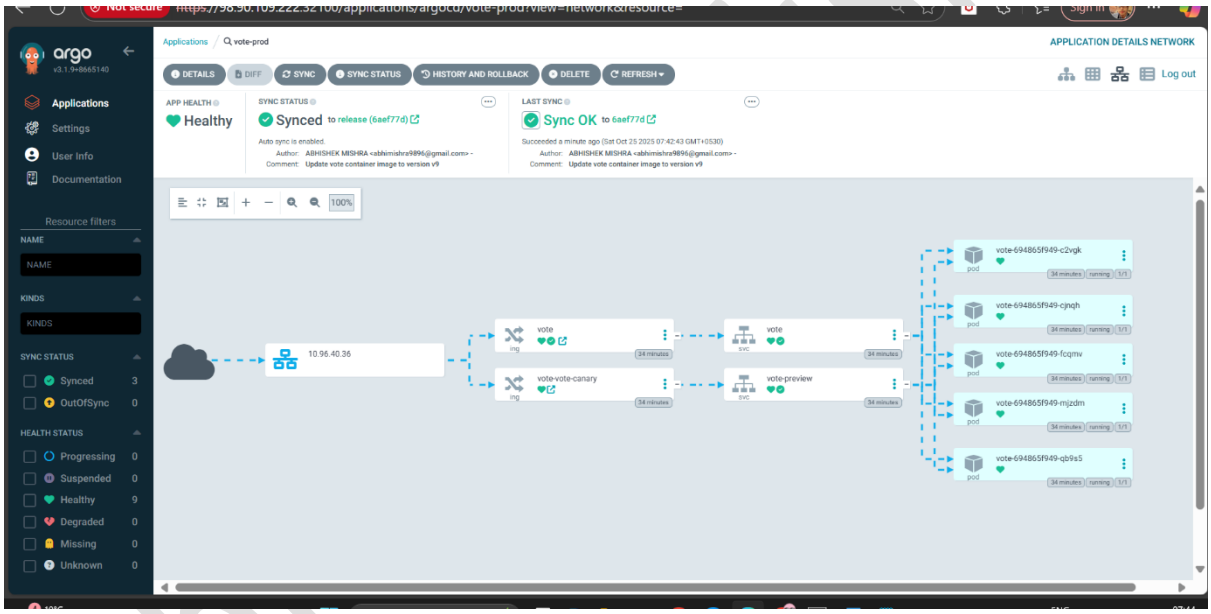

98.90.109.222 (ubuntu) (2)
root@ip-10-0-0-5:~/argo-labs# kubectl get application -A
NAMESPACE NAME SYNC STATUS HEALTH STATUS
argocd vote-staging Synced Healthy
root@ip-10-0-0-5:~/argo-labs#
Remote monitoring
Follow terminal folder

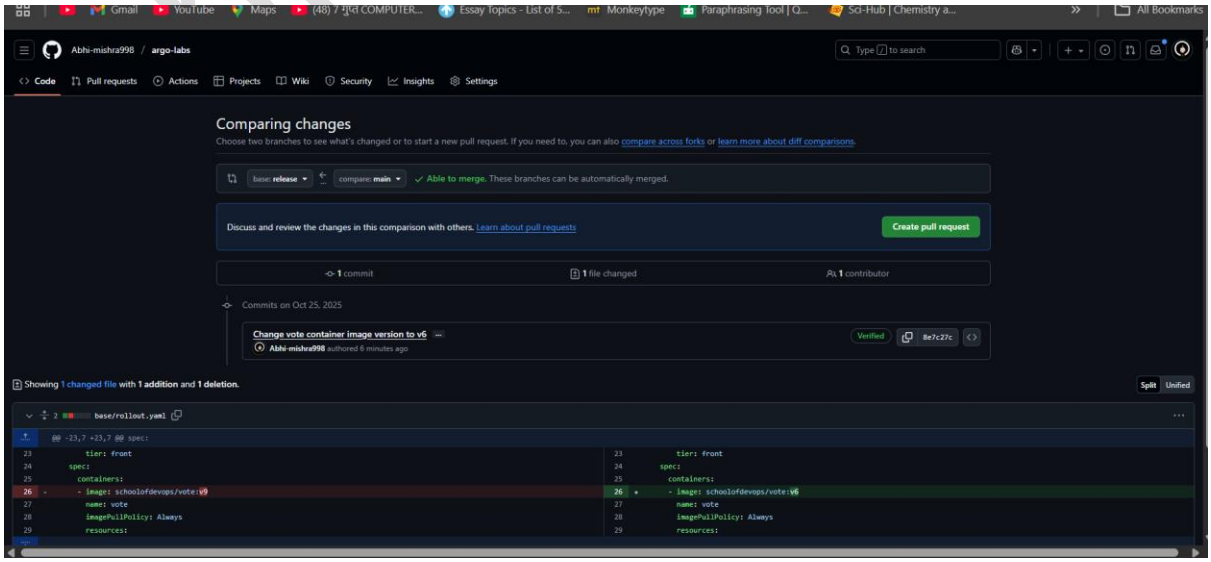

Not secure
https://98.90.109.222:32100/applications/argocd/vote-prod?view=network&resource=
Sign in
Applications
Settings
User Info
Documentation
APPLICATION DETAILS NETWORK
Log out
Healthy
OutOfSync
Sync OK
4 Warnings
10.96.40.36

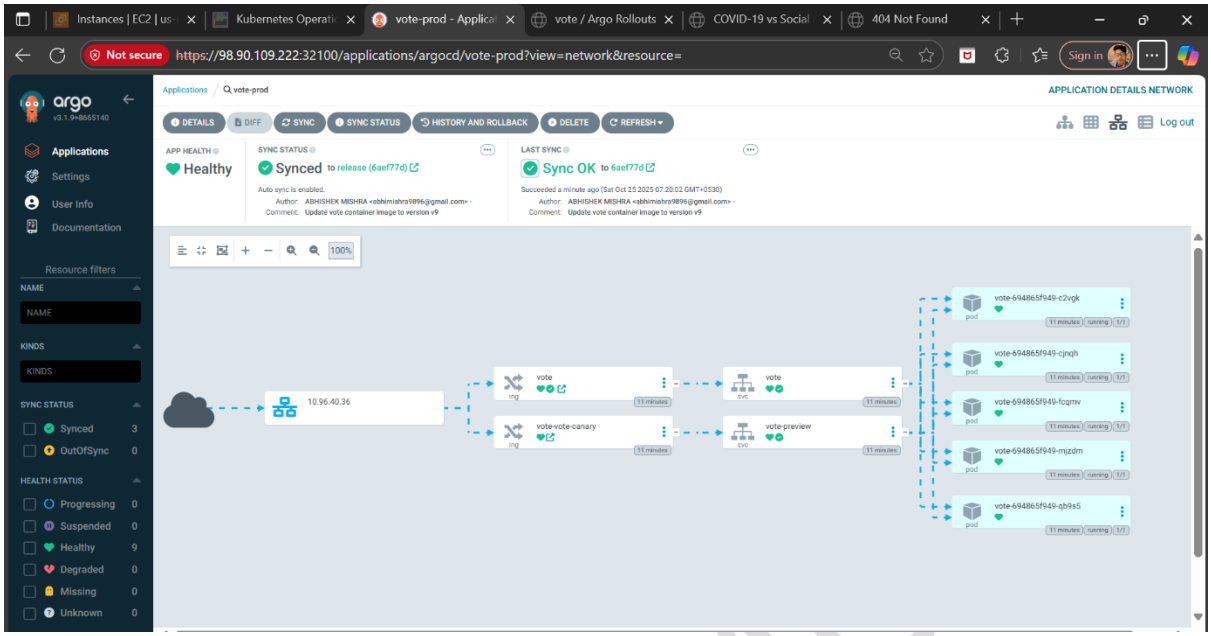

argo
prod
vote
Restart
Retry
Abort
Promote
PromoteFull
Steps
Canary Scale
Set Weight: 20%
Pause: 10s
Set Weight: 40%
Pause: 10s
Set Weight: 60%
Pause: 10s
Set Weight: 80%
Pause: 10s
Set Weight: 100%
Summary
Strategy
Canary
Step
10/10
Set Weight
100
Actual Weight
100
Containers
Edit
vote
schoolofdevops/votev6
Revisions
Revision 2
schoolofdevops/votev6
stable
vote-5f49849856
Revision 1
Rollback
schoolofdevops/votev9
vote-694865f949
00:23
min

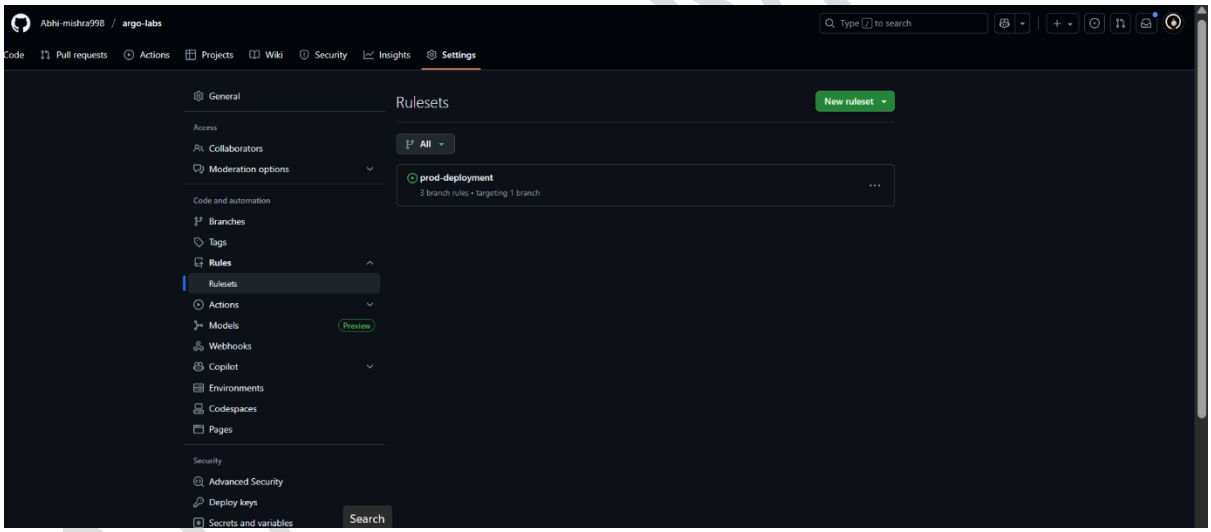

Not secure
98.90.109.222:3100/rollouts/rollout/prod/vote
Sign in
argo
prod
vote
Restart
Retry
Abort
Promote
PromoteFull
Steps
Canary Scale
Set Weight: 20%
Pause: 10s
Set Weight: 40%
Pause: 10s
Set Weight: 60%
Pause: 10s
Set Weight: 80%
Pause: 10s
Set Weight: 100%
Summary
Strategy
Canary
Step
8/10
Set Weight
80
Actual Weight
80
Containers
Edit
vote
schoolofdevops/votev6
Revisions
Revision 2
schoolofdevops/votev6
canary
vote-5f49849856
Revision 1
Rollback
schoolofdevops/votev9
stable
vote-694865f949

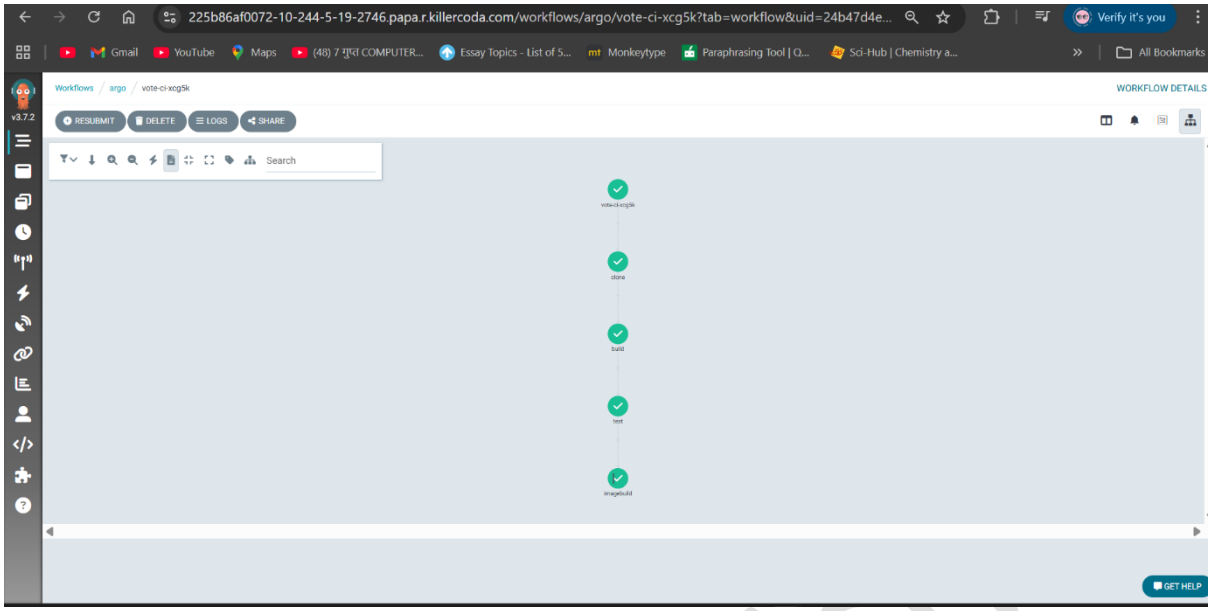

Not secure 98.90.109.222:3100/rollouts/rollout/prod/vote
Sign in
vote
Restart
Steps
Canary Scale
Set Weight: 20%
Pause: 10s
Set Weight: 40%
Pause: 10s
Set Weight: 60%
Pause: 10s
Set Weight: 80%
Pause: 10s
Set Weight: 100%
Summary
Strategy
Canary
Step
10/10
Set Weight
100
Actual Weight
100
Containers
Edit
vote
schoolofdevops/votev9
Revisions
Revision 1
schoolofdevops/votev9
stable
vote-694865f949

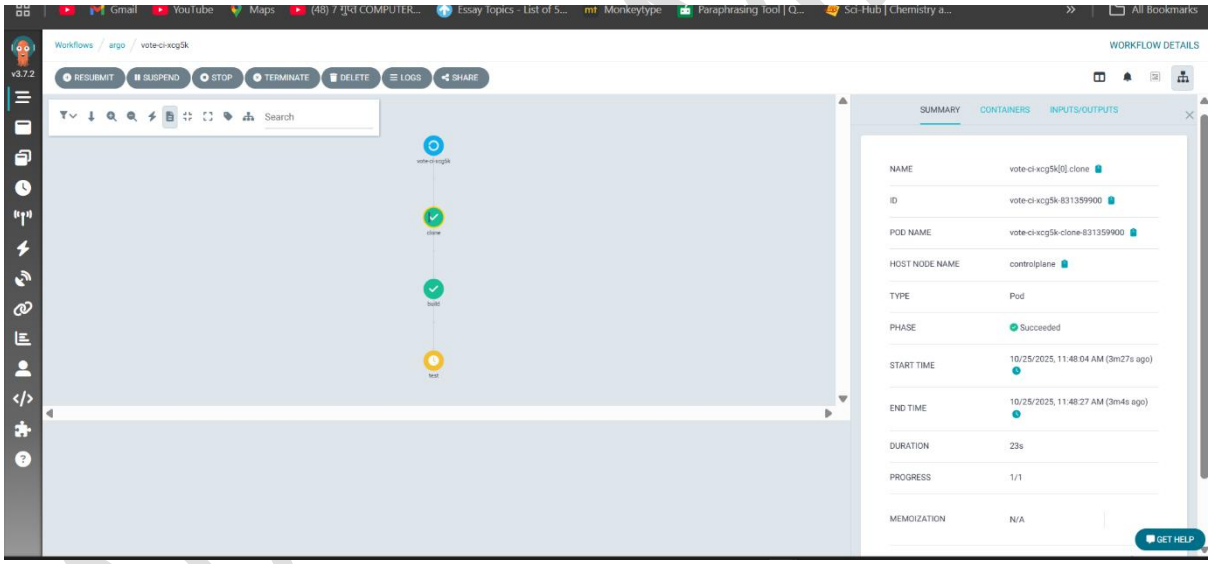

Applications
vote-prod
APPLICATION DETAILS NETWORK
DETAILS
DIFF
SYNC
SYNC STATUS
HISTORY AND ROLLBACK
DELETE
REFRESH
Log out
Applications
Settings
User Info
Documentation
Resource filters
APP HEALTH
Healthy
SYNC STATUS
Synced to release (6aef77d)
LAST SYNC
Sync OK to 6aef77d
Succeeded a minute ago (Sat Oct 25 2025 07:42:43 GMT+0530)
Author: ABHISHEK MISHRA <abhimishra9896@gmail.com> -
Comment: Update vote container image to version v9
NAME
KINDS
SYNC STATUS
Synced 3
OutOfSync 0
HEALTH STATUS
Progressing 0
Suspended 0
Healthy 9
Degraded 0
Missing 0
Unknown 0
10.96.40.36
vote
vote-vote-canary
vote-preview
vote-694865f949-c2vgk
vote-694865f949-cjnqh
vote-694865f949-fcqmv
vote-694865f949-mjzdm
vote-694865f949-qb9s5

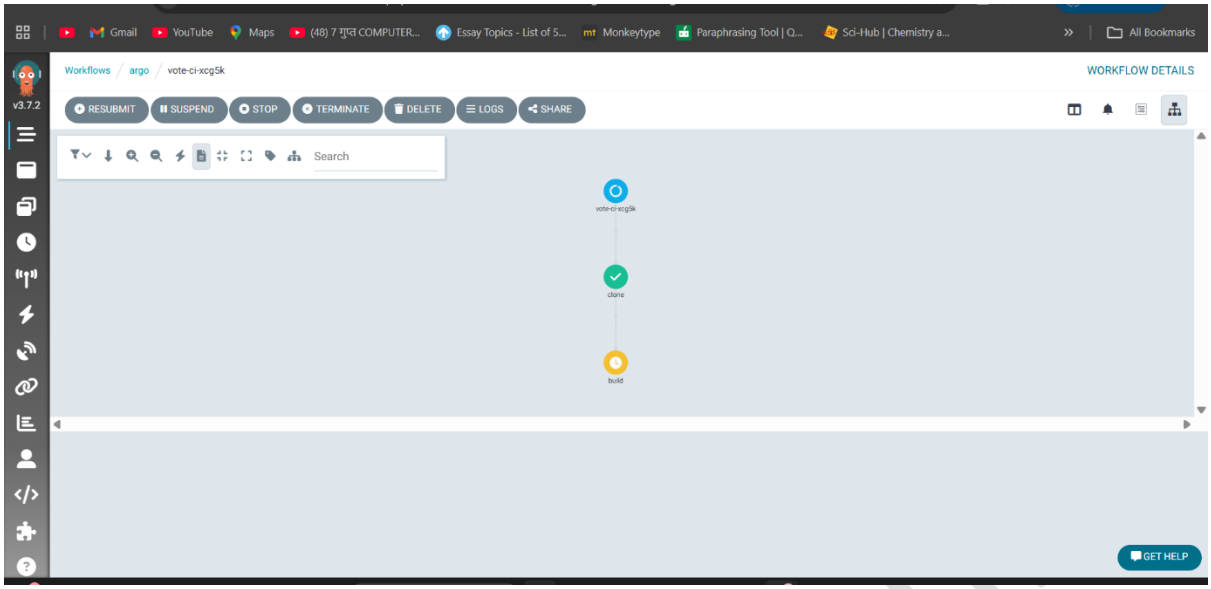

Abhi-mishra998 / argo-labs
Code
Pull requests
Actions
Projects
Wiki
Security
Insights
Settings
Comparing changes
Choose two branches to see what's changed or to start a new pull request. If you need to, you can also compare across forks or learn more about diff comparisons.
base: release
compare: main
Able to merge. These branches can be automatically merged.
Discuss and review the changes in this comparison with others. Learn about pull requests
Create pull request
1 commit
1 file changed
1 contributor
Commits on Oct 25, 2025
Change vote container image version to v6
Abhi-mishra998 authored 6 minutes ago
Verified
8e7c27c
Showing 1 changed file with 1 addition and 1 deletion.
Split
Unified
base/rollout.yaml
@@ -23,7 +23,7 @@ spec:
tier: front
spec:
containers:
- image: schoolofdevops/vote:v9
name: vote
imagePullPolicy: Always
resources:
tier: front
spec:
containers:
- image: schoolofdevops/vote:v6
name: vote
imagePullPolicy: Always
resources:

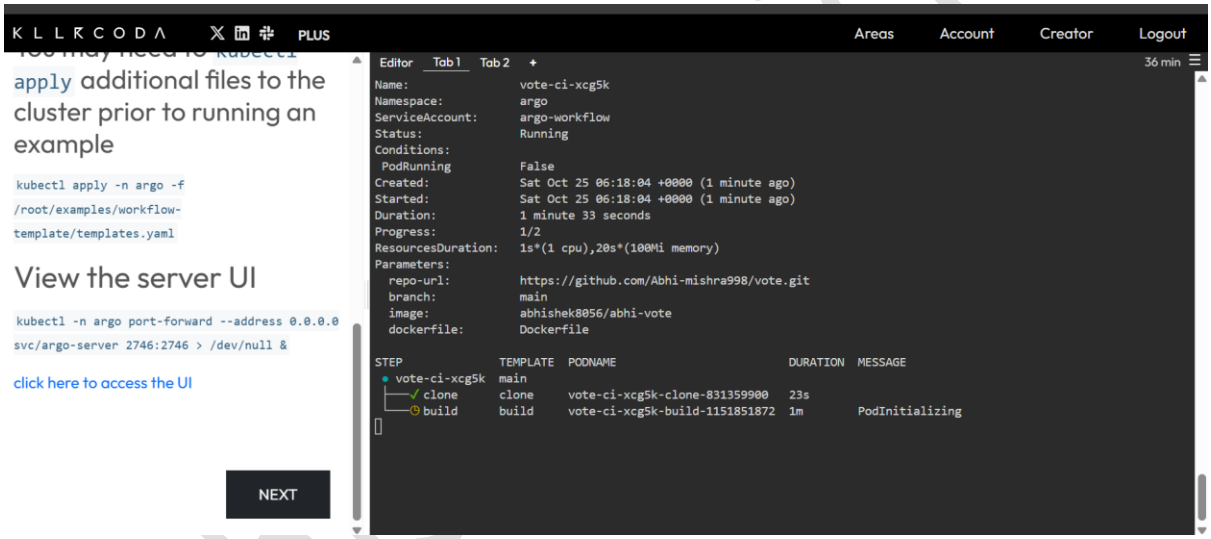

Not secure
https://98.90.109.222:32100/applications/argocd/vote-prod?view=network&resource=
argo
Applications
Settings
User Info
Documentation
Resource filters
NAME
KINDS
SYNC STATUS
Synced 3
OutOfSync 0
HEALTH STATUS
Progressing 0
Suspended 0
Healthy 9
Degraded 0
Missing 0
Unknown 0
Applications / vote-prod
APPLICATION DETAILS NETWORK
DETAILS
DIFF
SYNC
SYNC STATUS
HISTORY AND ROLLBACK
DELETE
REFRESH
Log out
APP HEALTH
Healthy
SYNC STATUS
Synced to release (6aef77d)
LAST SYNC
Sync OK to 6aef77d
10.96.40.36
vote
vote-vote-canary
vote-preview
vote-694865f949-c2vgk
vote-694865f949-cjnqh
vote-694865f949-fcqmv
vote-694865f949-mjzdm
vote-694865f949-qb9s5

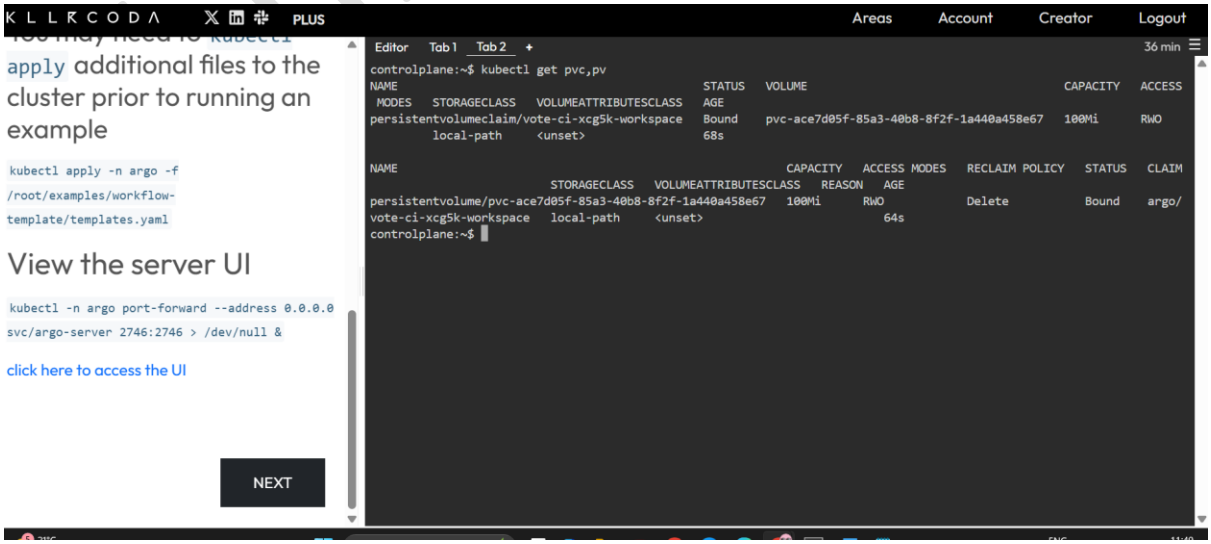

Abhi-mishra998 / argo-labs
Code
Pull requests
Actions
Projects
Wiki
Security
Insights
Settings
General
Access
Collaborators
Moderation options
Code and automation
Branches
Tags
Rules
Rulesets
Actions
Models
Preview
Webhooks
Copilot
Environments
Codespaces
Pages
Security
Advanced Security
Deploy keys
Secrets and variables
Rulesets
New ruleset
All
prod-deployment
3 branch rules • targeting 1 branch
Search

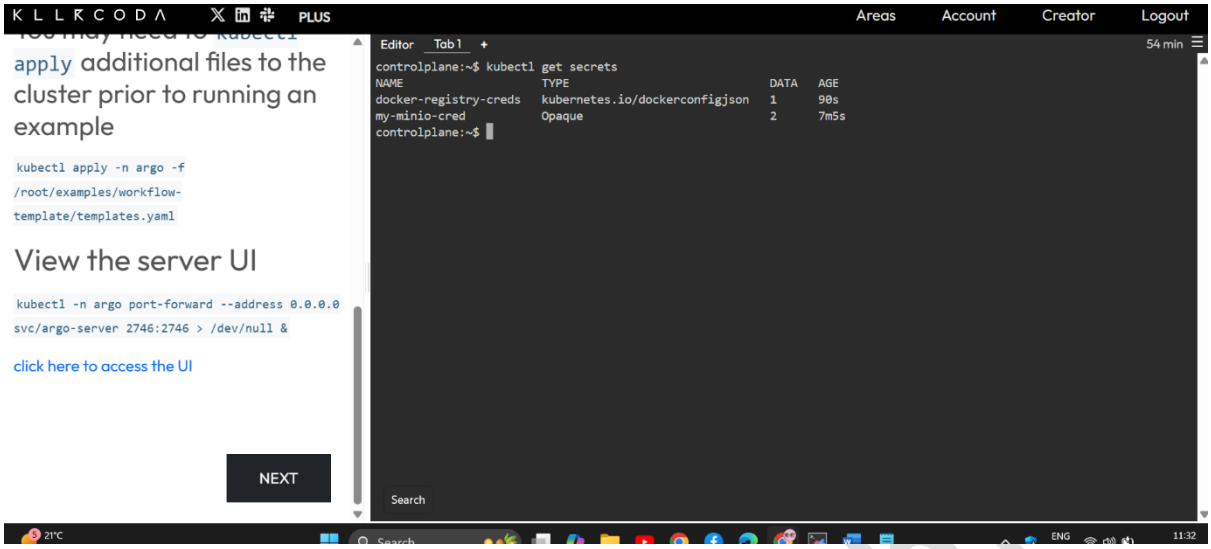

225b86af0072-10-244-5-19-2746.papa.r.killercoda.com/workflows/argo/vote-ci-xcg5k?tab=workflow&uid=24b47d4e...
Verify it's you
Gmail
YouTube
Maps
Essay Topics - List of 5...
Monkeytype
Paraphrasing Tool | Q...
Sci-Hub | Chemistry a...
All Bookmarks
Workflows
argo
vote-ci-xcg5k
WORKFLOW DETAILS
RESUBMIT
DELETE
LOGS
SHARE
Search
vote-ci-xcg5k
clone
build
test
imagebuild
GET HELP

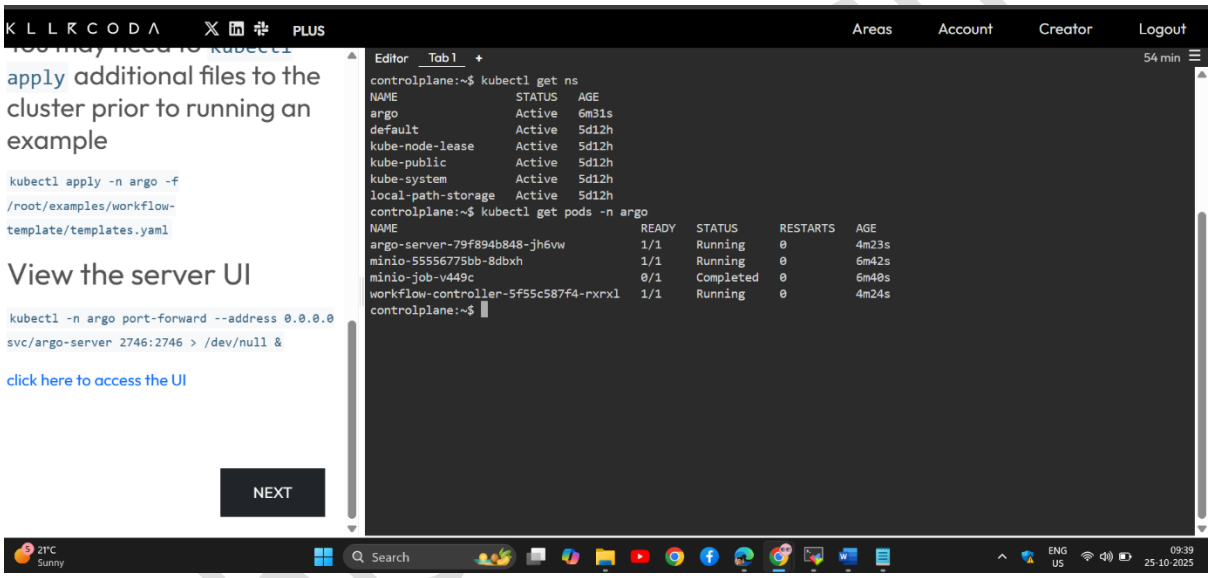

Workflows
argo
vote-ci-xcg5k
WORKFLOW DETAILS
RESUBMIT
SUSPEND
STOP
TERMINATE
DELETE
LOGS
SHARE
Search
SUMMARY
CONTAINERS
INPUTS/OUTPUTS
NAME
vote-ci-xcg5k[0].clone
ID
vote-ci-xcg5k-831359900
POD NAME
vote-ci-xcg5k-clone-831359900
HOST NODE NAME
controlplane
TYPE
Pod
PHASE
Succeeded
START TIME
10/25/2025, 11:48:04 AM (3m27s ago)
END TIME
10/25/2025, 11:48:27 AM (3m4s ago)
DURATION
23s
PROGRESS
1/1
MEMOIZATION
N/A
GET HELP

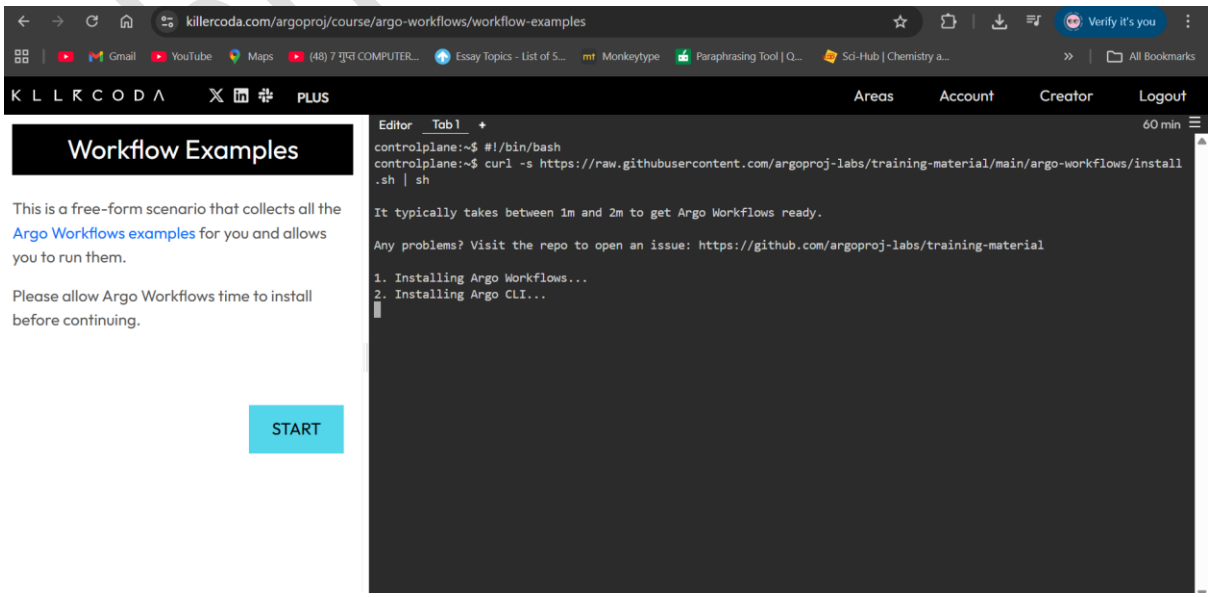

Gmail
YouTube
Maps
Monkeytype
Paraphrasing Tool | Q...
Sci-Hub | Chemistry a...
All Bookmarks
Workflows / argo / vote-ci-xcg5k
WORKFLOW DETAILS
v3.7.2
RESUBMIT
SUSPEND
STOP
TERMINATE
DELETE
LOGS
SHARE
Search
vote-ci-xcg5k
clone
build
GET HELP

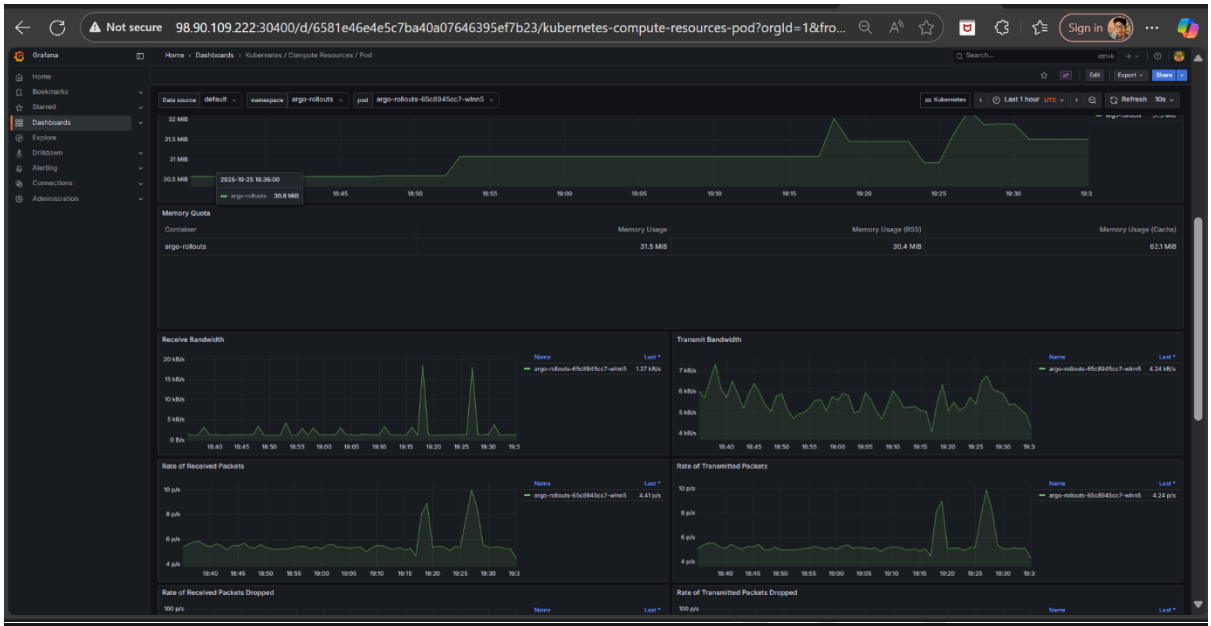

KLLKCODA
PLUS
Areas
Account
Creator
Logout
apply additional files to the cluster prior to running an example
kubectl apply -n argo -f /root/examples/workflow-template/templates.yaml
View the server UI
kubectl -n argo port-forward --address 0.0.0.0 svc/argo-server 2746:2746 > /dev/null &
click here to access the UI
NEXT
Editor
Tab 1
Tab 2
36 min
Name: vote-ci-xcg5k
Namespace: argo
ServiceAccount: argo-workflow
Status: Running
Conditions:
PodRunning False
Created: Sat Oct 25 06:18:04 +0000 (1 minute ago)
Started: Sat Oct 25 06:18:04 +0000 (1 minute ago)
Duration: 1 minute 33 seconds
Progress: 1/2
ResourcesDuration: 1s*(1 cpu),20s*(100Mi memory)
Parameters:
repo-url: https://github.com/Abhi-mishra998/vote.git
branch: main
image: abhishek8056/abhi-vote
dockerfile: Dockerfile
STEP TEMPLATE PODNAME DURATION MESSAGE
vote-ci-xcg5k main
clone clone vote-ci-xcg5k-clone-831359900 23s
build build vote-ci-xcg5k-build-1151851872 1m PodInitializing

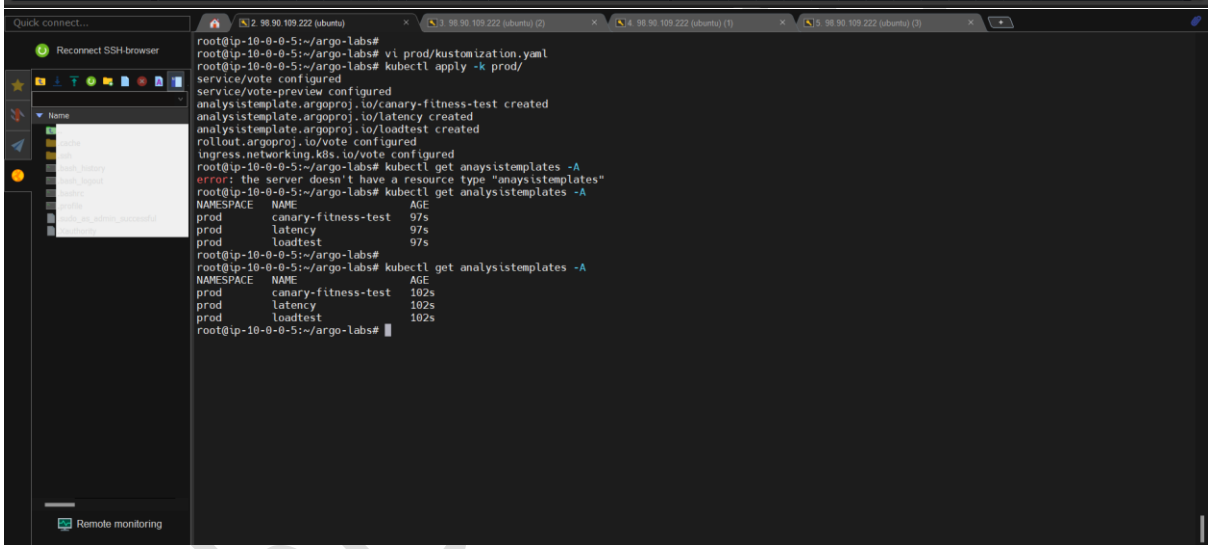

KLLKCODA
PLUS
Areas
Account
Creator
Logout
apply additional files to the cluster prior to running an example
kubectl apply -n argo -f /root/examples/workflow-template/templates.yaml
View the server UI
kubectl -n argo port-forward --address 0.0.0.0 svc/argo-server 2746:2746 > /dev/null &
click here to access the UI
NEXT
Editor
Tab 1
Tab 2
36 min
controlplane:~$ kubectl get pvc,pv
NAME STATUS VOLUME CAPACITY ACCESS MODES STORAGECLASS VOLUMEATTRIBUTESCLASS AGE
persistentvolumeclaim/vote-ci-xcg5k-workspace Bound pvc-ace7d05f-85a3-40b8-8f2f-1a440a458e67 100Mi RWO local-path <unset> 68s
NAME CAPACITY ACCESS MODES RECLAIM POLICY STATUS CLAIM STORAGECLASS VOLUMEATTRIBUTESCLASS REASON AGE
persistentvolume/pvc-ace7d05f-85a3-40b8-8f2f-1a440a458e67 100Mi RWO Delete Bound argo/vote-ci-xcg5k-workspace local-path <unset> 64s
controlplane:~$

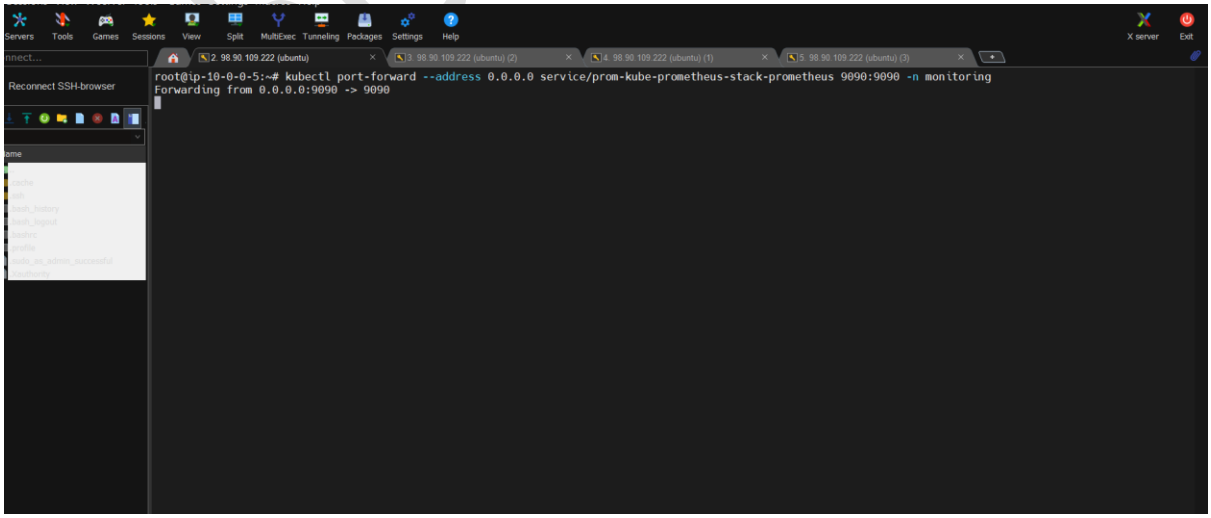

KLLRCODA
PLUS
Areas
Account
Creator
Logout
apply additional files to the cluster prior to running an example
kubectl apply -n argo -f /root/examples/workflow-template/templates.yaml
View the server UI
kubectl -n argo port-forward --address 0.0.0.0 svc/argo-server 2746:2746 > /dev/null &
click here to access the UI
NEXT
Editor
Tab 1
54 min
controlplane:~$ kubectl get secrets
NAME TYPE DATA AGE
docker-registry-creds kubernetes.io/dockerconfigjson 1 90s
my-minio-cred Opaque 2 7m5s
controlplane:~$
Search

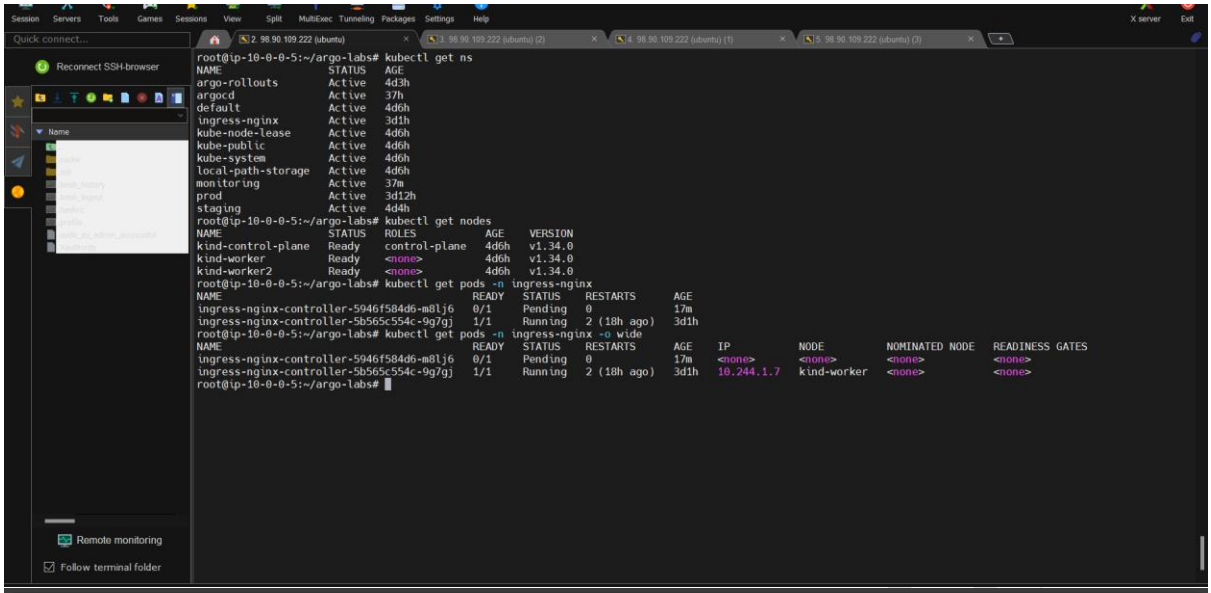

KLLRCODA
PLUS
Areas
Account
Creator
Logout
apply additional files to the cluster prior to running an example
kubectl apply -n argo -f /root/examples/workflow-template/templates.yaml
View the server UI
kubectl -n argo port-forward --address 0.0.0.0 svc/argo-server 2746:2746 > /dev/null &
click here to access the UI
NEXT
Editor
Tab 1
54 min
controlplane:~$ kubectl get ns
NAME STATUS AGE
argo Active 6m31s
default Active 5d12h
kube-node-lease Active 5d12h
kube-public Active 5d12h
kube-system Active 5d12h
local-path-storage Active 5d12h
controlplane:~$ kubectl get pods -n argo
NAME READY STATUS RESTARTS AGE
argo-server-79f894b848-jh6vw 1/1 Running 0 4m23s
minio-55556775bb-8dbxh 1/1 Running 0 6m42s
minio-job-v449c 0/1 Completed 0 6m40s
workflow-controller-5f55c587f4-rxrxl 1/1 Running 0 4m24s
controlplane:~$
Search

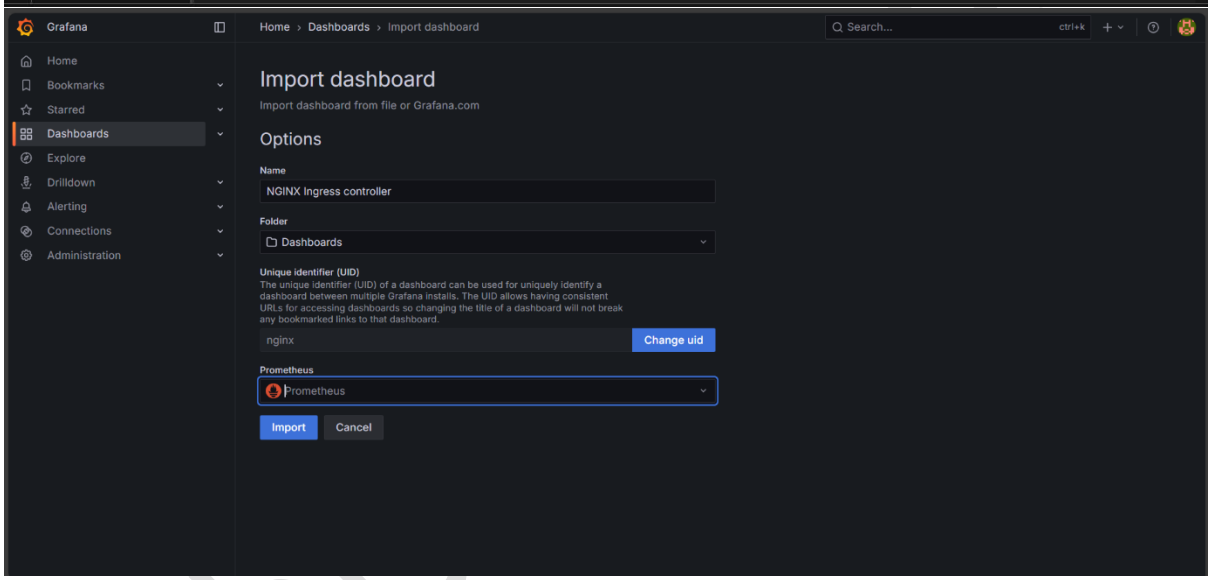

killercoda.com/argoproj/course/argo-workflows/workflow-examples
KLLRCODA
PLUS
Areas
Account
Creator
Logout
Workflow Examples
This is a free-form scenario that collects all the Argo Workflows examples for you and allows you to run them.
Please allow Argo Workflows time to install before continuing.
START
Editor
Tab 1
60 min
controlplane:~$ #!/bin/bash
controlplane:~$ curl -s https://raw.githubusercontent.com/argoproj-labs/training-material/main/argo-workflows/install.sh | sh
It typically takes between 1m and 2m to get Argo Workflows ready.
Any problems? Visit the repo to open an issue: https://github.com/argoproj-labs/training-material
1. Installing Argo Workflows...
2. Installing Argo CLI...

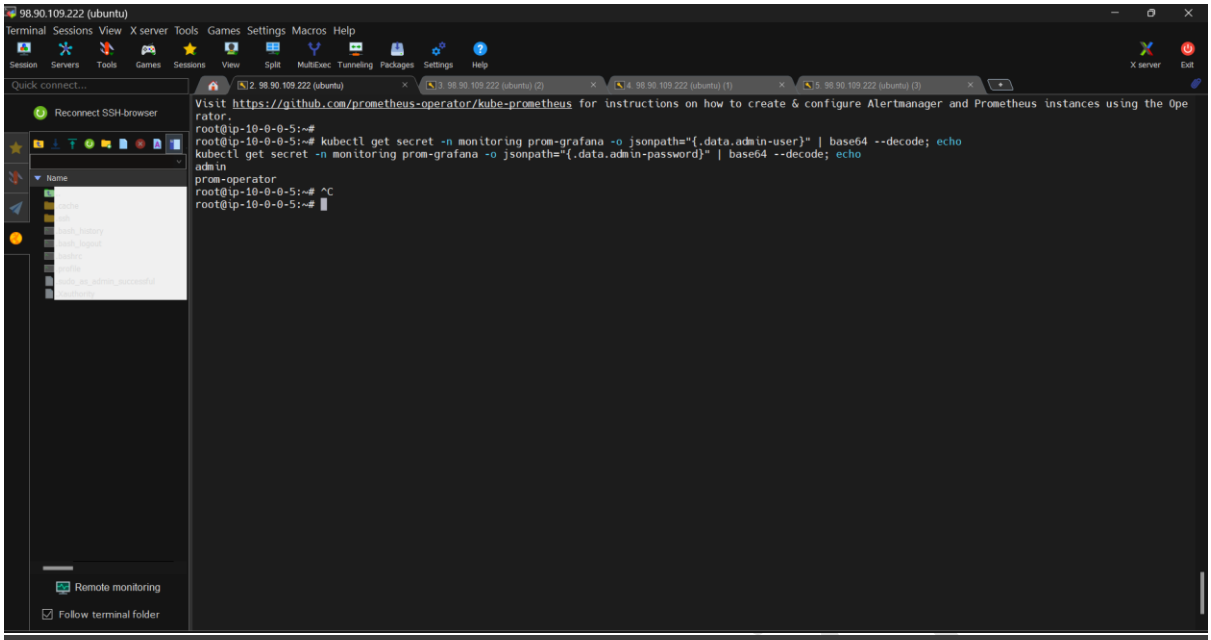

Not secure 98.90.109.222:30400/d/6581e46e4e5c7ba40a07646395ef7b23/kubernetes-compute-resources-pod?orgId=1&fro...
Sign in
Grafana
Home › Dashboards › Kubernetes / Compute Resources / Pod
Memory Quota
Container
Memory Usage
Memory Usage (RSS)
Memory Usage (Cache)
argo-rollouts
31.5 MiB
30.4 MiB
62.1 MiB
Receive Bandwidth
Transmit Bandwidth
Rate of Received Packets
Rate of Transmitted Packets
Rate of Received Packets Dropped
Rate of Transmitted Packets Dropped

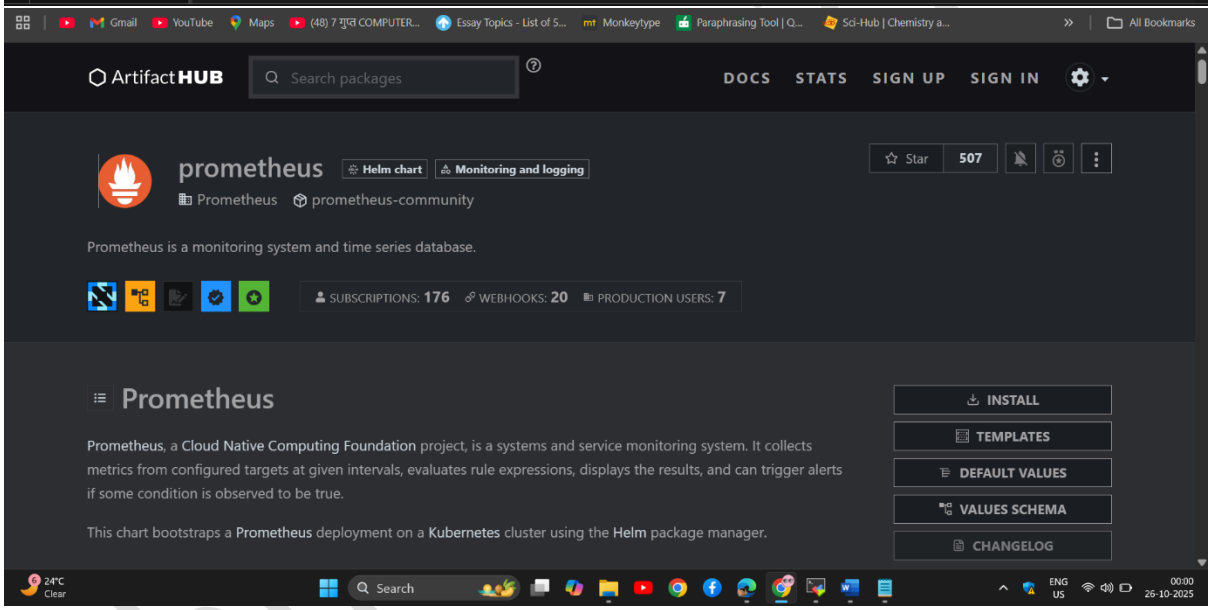

Quick connect...
Reconnect SSH-browser
Remote monitoring
root@ip-10-0-0-5:~/argo-labs#
root@ip-10-0-0-5:~/argo-labs# vi prod/kustomization.yaml
root@ip-10-0-0-5:~/argo-labs# kubectl apply -k prod/
service/vote configured
service/vote-preview configured
analysistemplate.argoproj.io/canary-fitness-test created
analysistemplate.argoproj.io/latency created
analysistemplate.argoproj.io/loadtest created
rollout.argoproj.io/vote configured
ingress.networking.k8s.io/vote configured
root@ip-10-0-0-5:~/argo-labs# kubectl get anaysistemplates -A
error: the server doesn't have a resource type "anaysistemplates"
root@ip-10-0-0-5:~/argo-labs# kubectl get analysistemplates -A
NAMESPACE NAME AGE
prod canary-fitness-test 97s
prod latency 97s
prod loadtest 97s
root@ip-10-0-0-5:~/argo-labs#
root@ip-10-0-0-5:~/argo-labs# kubectl get analysistemplates -A
NAMESPACE NAME AGE
prod canary-fitness-test 102s
prod latency 102s
prod loadtest 102s
root@ip-10-0-0-5:~/argo-labs#

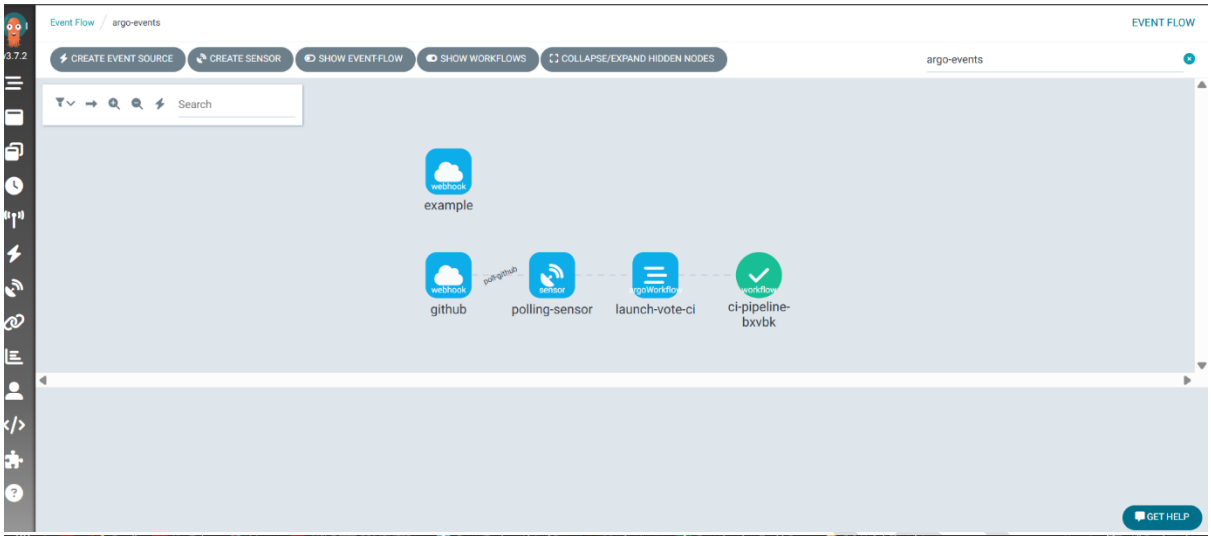

Servers
Tools
Games
Sessions
View
Split
MultiExec
Tunneling
Packages
Settings
Help
X server
Exit
Reconnect SSH-browser
root@ip-10-0-0-5:~# kubectl port-forward --address 0.0.0.0 service/prom-kube-prometheus-stack-prometheus 9090:9090 -n monitoring
Forwarding from 0.0.0.0:9090 -> 9090

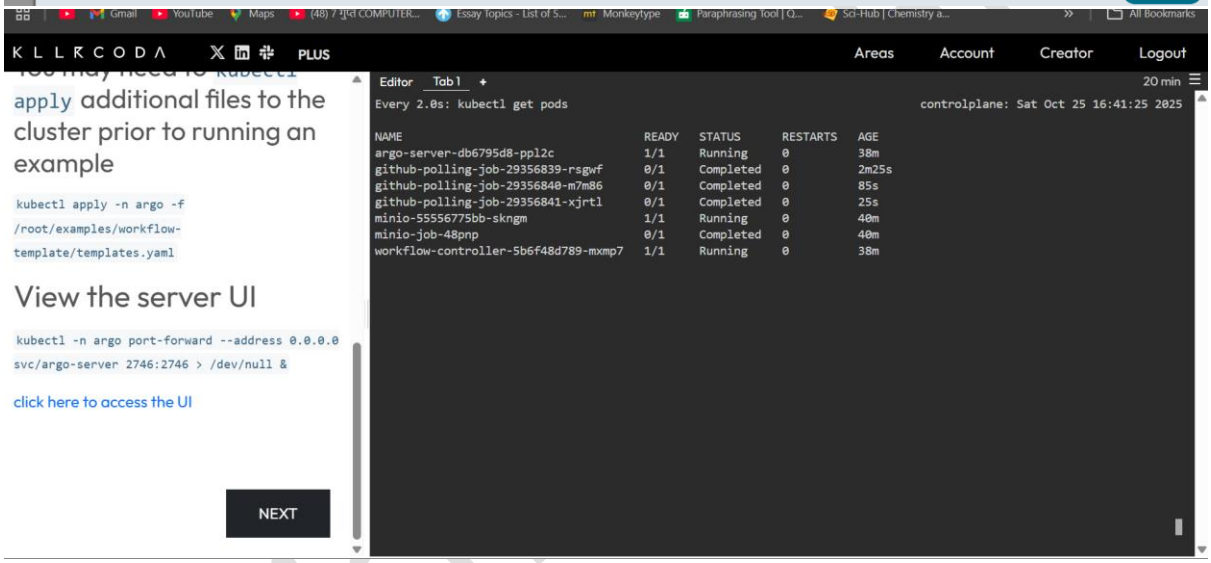

Session Servers Tools Games Sessions View Split MultiExec Tunneling Packages Settings Help X server Exit
Quick connect...
Reconnect SSH-browser
Name
Remote monitoring
Follow terminal folder
root@ip-10-0-0-5:~/argo-labs# kubectl get ns
NAME STATUS AGE
argo-rollouts Active 4d3h
argocd Active 37h
default Active 4d6h
ingress-nginx Active 3d1h
kube-node-lease Active 4d6h
kube-public Active 4d6h
kube-system Active 4d6h
local-path-storage Active 4d6h
monitoring Active 37m
prod Active 3d12h
staging Active 4d4h
root@ip-10-0-0-5:~/argo-labs# kubectl get nodes
NAME STATUS ROLES AGE VERSION
kind-control-plane Ready control-plane 4d6h v1.34.0
kind-worker Ready <none> 4d6h v1.34.0
kind-worker2 Ready <none> 4d6h v1.34.0
root@ip-10-0-0-5:~/argo-labs# kubectl get pods -n ingress-nginx
NAME READY STATUS RESTARTS AGE
ingress-nginx-controller-5946f584d6-m8lj6 0/1 Pending 0 17m
ingress-nginx-controller-5b565c554c-9g7gj 1/1 Running 2 (18h ago) 3d1h
root@ip-10-0-0-5:~/argo-labs# kubectl get pods -n ingress-nginx -o wide
NAME READY STATUS RESTARTS AGE IP NODE NOMINATED NODE READINESS GATES
ingress-nginx-controller-5946f584d6-m8lj6 0/1 Pending 0 17m <none> <none> <none> <none>
ingress-nginx-controller-5b565c554c-9g7gj 1/1 Running 2 (18h ago) 3d1h 10.244.1.7 kind-worker <none> <none>
root@ip-10-0-0-5:~/argo-labs#

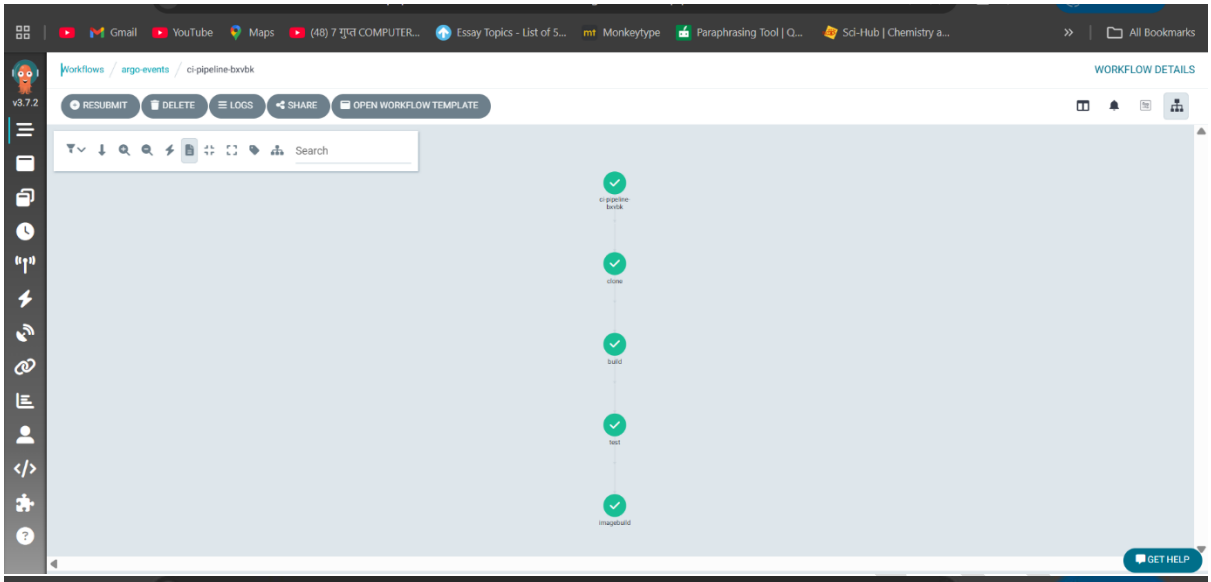

Grafana
Home
Bookmarks
Starred
Dashboards
Explore
Drilldown
Alerting
Connections
Administration
Home › Dashboards › Import dashboard
Search... ctrl+k
Import dashboard
Import dashboard from file or Grafana.com
Options
Name
NGINX Ingress controller
Folder
Dashboards
Unique identifier (UID)
The unique identifier (UID) of a dashboard can be used for uniquely identify a dashboard between multiple Grafana installs. The UID allows having consistent URLs for accessing dashboards so changing the title of a dashboard will not break any bookmarked links to that dashboard.
nginx
Change uid
Prometheus
Prometheus
Import
Cancel

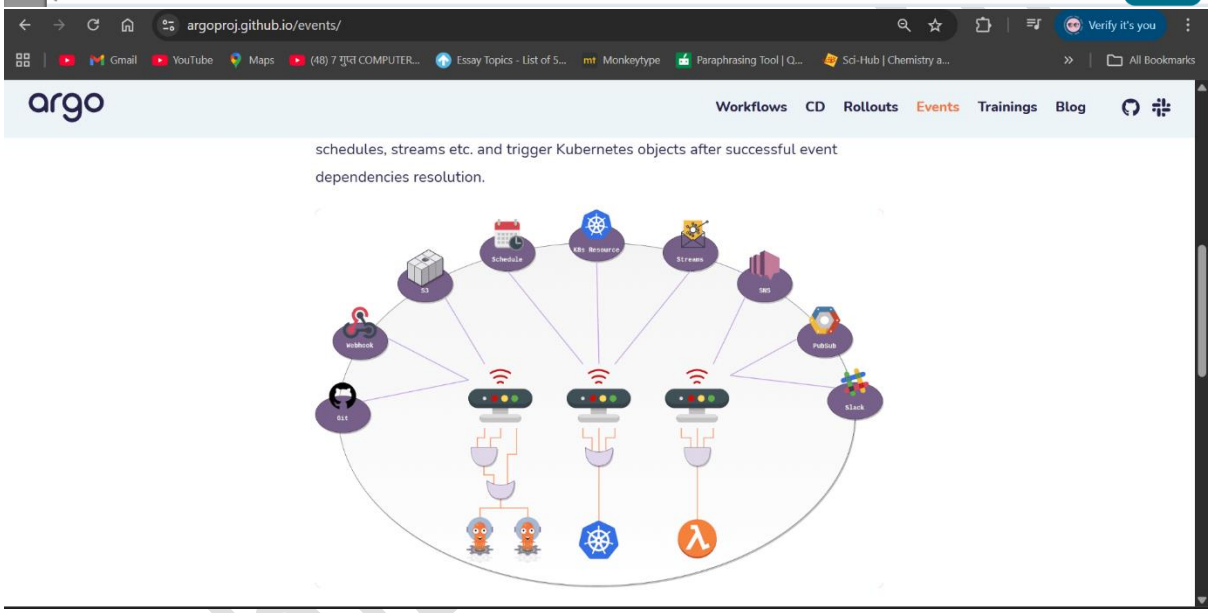

98.90.109.222 (ubuntu)
Terminal Sessions View X server Tools Games Settings Macros Help
Session Servers Tools Games Sessions View Split MultiExec Tunneling Packages Settings Help
X server Exit
Quick connect...
Reconnect SSH-browser
2. 98.90.109.222 (ubuntu)
Visit https://github.com/prometheus-operator/kube-prometheus for instructions on how to create & configure Alertmanager and Prometheus instances using the Operator.
root@ip-10-0-0-5:~#
root@ip-10-0-0-5:~# kubectl get secret -n monitoring prom-grafana -o jsonpath="{.data.admin-user}" | base64 --decode; echo
kubectl get secret -n monitoring prom-grafana -o jsonpath="{.data.admin-password}" | base64 --decode; echo
admin
prom-operator
root@ip-10-0-0-5:~# ^C
root@ip-10-0-0-5:~#
Name
Remote monitoring
Follow terminal folder

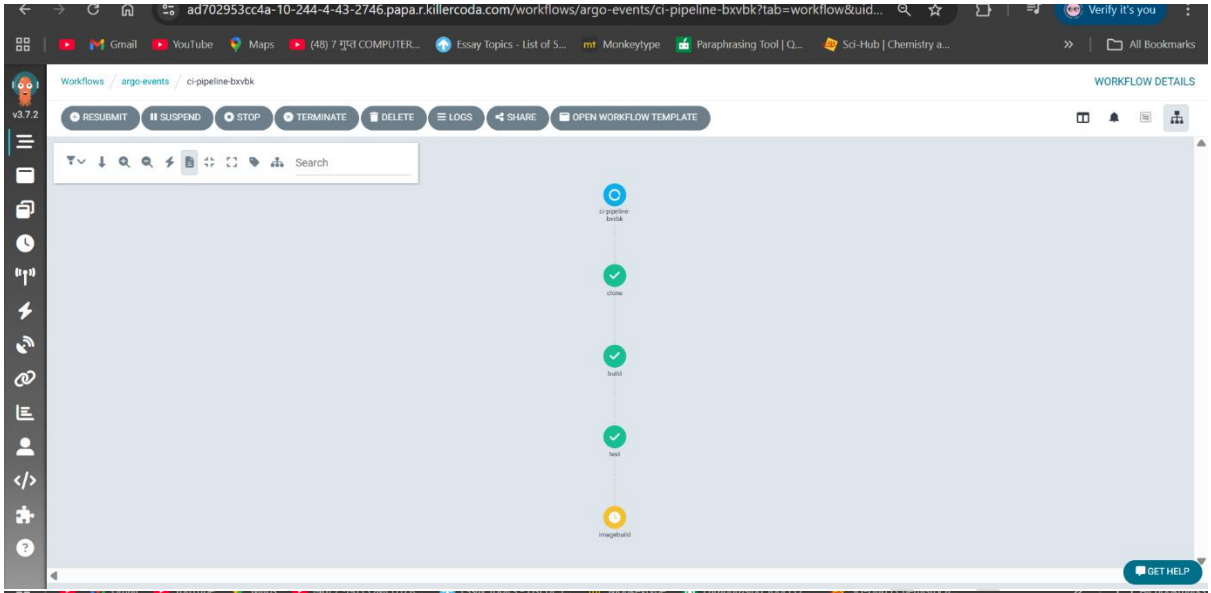

Gmail YouTube Maps (48) 7 गुप्त COMPUTER... Essay Topics - List of 5... Monkeytype Paraphrasing Tool | Q... Sci-Hub | Chemistry a... All Bookmarks
ArtifactHUB
Search packages
DOCS STATS SIGN UP SIGN IN
prometheus
Helm chart
Monitoring and logging
Star 507
Prometheus prometheus-community
Prometheus is a monitoring system and time series database.
SUBSCRIPTIONS: 176 WEBHOOKS: 20 PRODUCTION USERS: 7
Prometheus
Prometheus, a Cloud Native Computing Foundation project, is a systems and service monitoring system. It collects metrics from configured targets at given intervals, evaluates rule expressions, displays the results, and can trigger alerts if some condition is observed to be true.
This chart bootstraps a Prometheus deployment on a Kubernetes cluster using the Helm package manager.
INSTALL
TEMPLATES
DEFAULT VALUES
VALUES SCHEMA
CHANGELOG
24°C Clear
Search
ENG US
00:00 26-10-2025

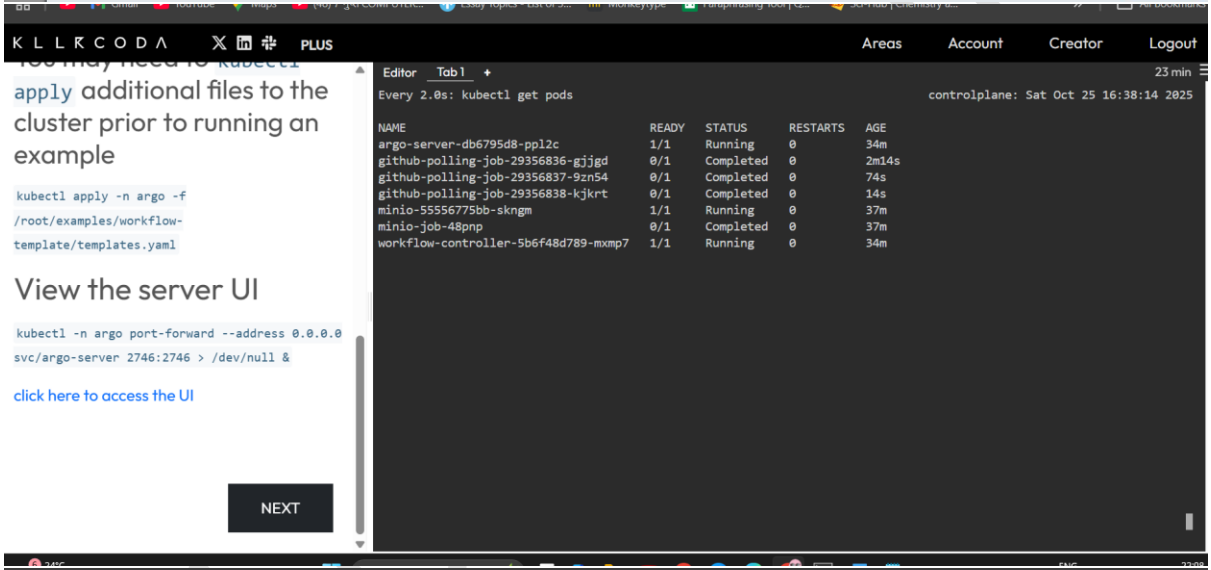

Event Flow
argo-events
EVENT FLOW
CREATE EVENT SOURCE
CREATE SENSOR
SHOW EVENT-FLOW
SHOW WORKFLOWS
COLLAPSE/EXPAND HIDDEN NODES
argo-events
Search
webhook
example
webhook
github
poll-github
sensor
polling-sensor
argoWorkflow
launch-vote-ci
workflow
ci-pipeline-bxvbk
GET HELP

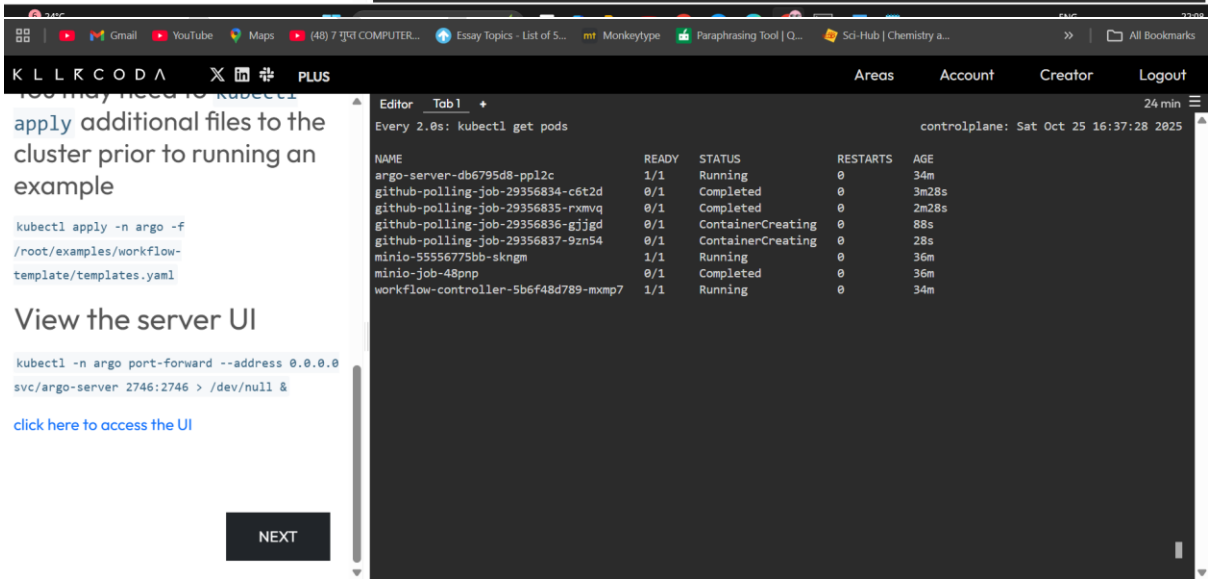

KLLRCODA
PLUS
Areas
Account
Creator
Logout
apply additional files to the cluster prior to running an example
kubectl apply -n argo -f /root/examples/workflow-template/templates.yaml
View the server UI
kubectl -n argo port-forward --address 0.0.0.0 svc/argo-server 2746:2746 > /dev/null &
click here to access the UI
NEXT
Editor
Tab 1
20 min
Every 2.0s: kubectl get pods
controlplane: Sat Oct 25 16:41:25 2025
NAME READY STATUS RESTARTS AGE
argo-server-db6795d8-ppl2c 1/1 Running 0 38m
github-polling-job-29356839-rsgwf 0/1 Completed 0 2m25s
github-polling-job-29356840-m7m86 0/1 Completed 0 85s
github-polling-job-29356841-xjrtl 0/1 Completed 0 25s
minio-55556775bb-skngm 1/1 Running 0 40m
minio-job-48pnp 0/1 Completed 0 40m
workflow-controller-5b6f48d789-mxmp7 1/1 Running 0 38m

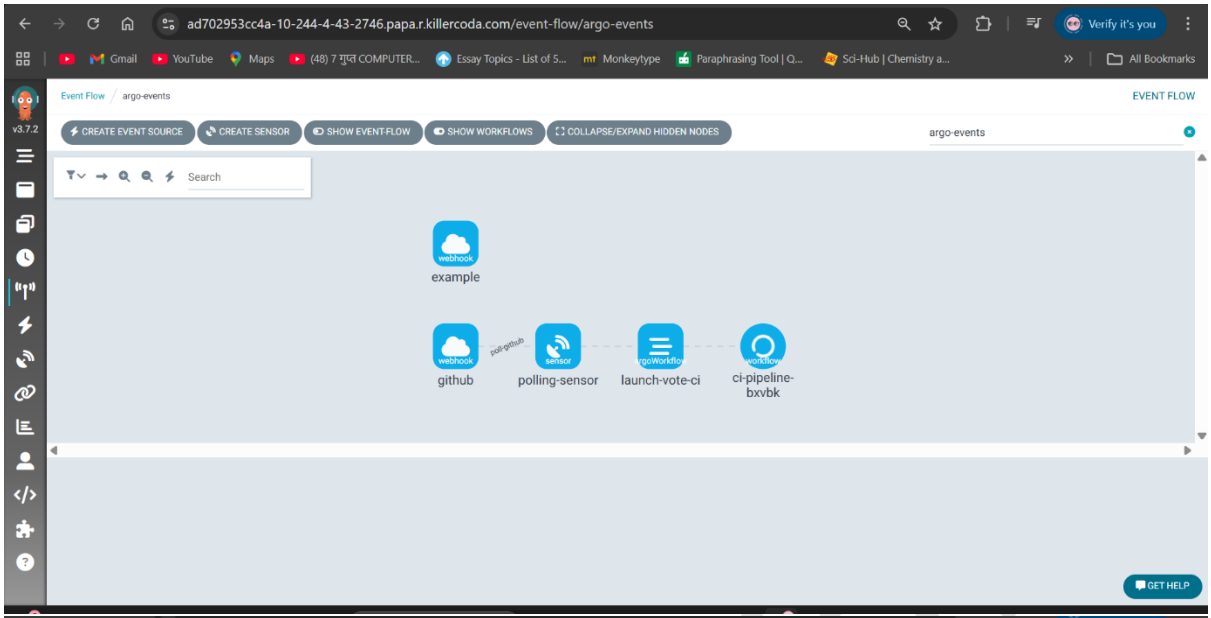

Workflows
argo-events
ci-pipeline-bxvbk
WORKFLOW DETAILS
RESUBMIT
DELETE
LOGS
SHARE
OPEN WORKFLOW TEMPLATE
Search
GET HELP

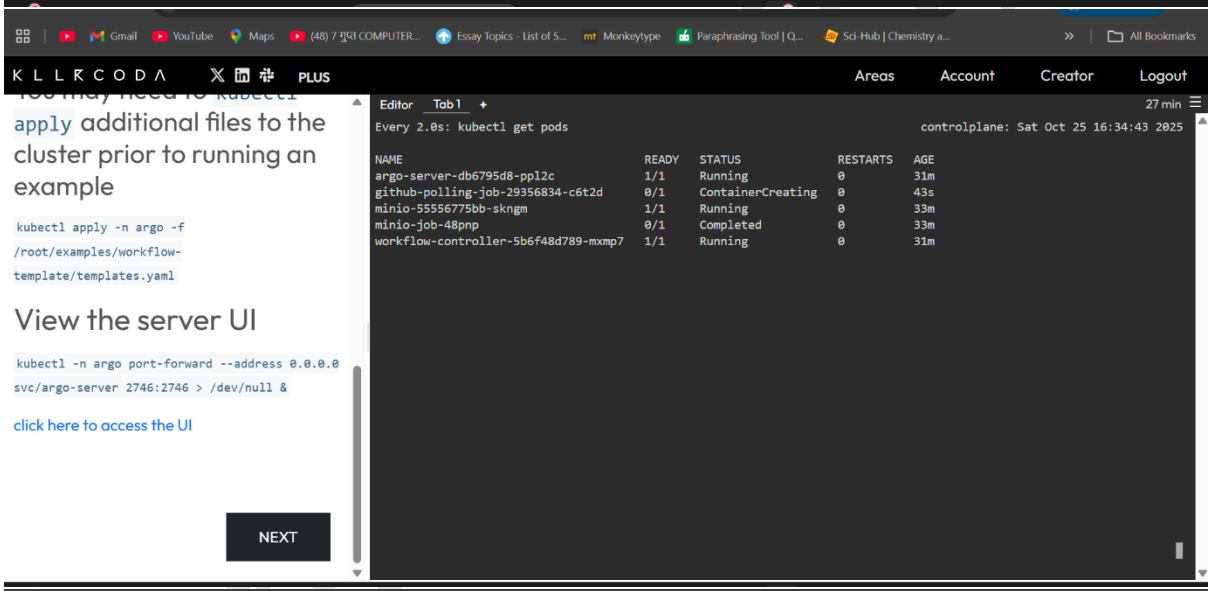

argoproj.github.io/events/
argo
Workflows
CD
Rollouts
Events
Trainings
Blog
schedules, streams etc. and trigger Kubernetes objects after successful event dependencies resolution.
Schedule
K8s Resource
Streams
S3
SNS
Webhook
PubSub
Git
Slack

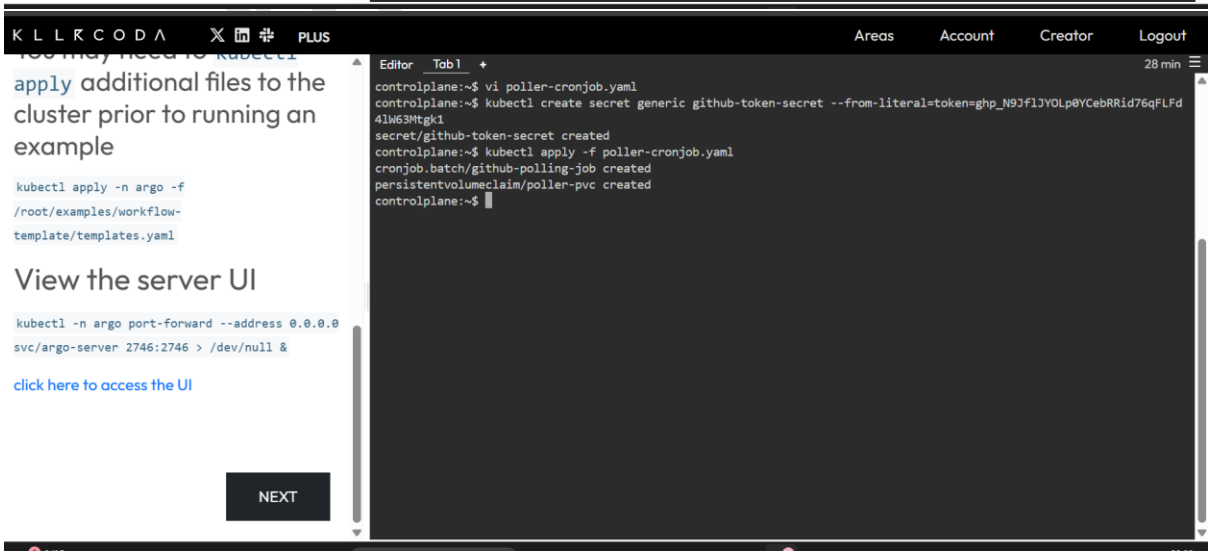

ad702953cc4a-10-244-4-43-2746.papa.r.killercoda.com/workflows/argo-events/ci-pipeline-bxvbk?tab=workflow&uid...
Verify it's you
Gmail
YouTube
Maps
(48) 7 मुफ्त COMPUTER...
Essay Topics - List of 5...
Monkeytype
Paraphrasing Tool | Q...
Sci-Hub | Chemistry a...
All Bookmarks
v3.7.2
Workflows
argo-events
ci-pipeline-bxvbk
WORKFLOW DETAILS
RESUBMIT
SUSPEND
STOP
TERMINATE
DELETE
LOGS
SHARE
OPEN WORKFLOW TEMPLATE
Search
ci-pipeline-bxvbk
clone
build
test
imagebuild
GET HELP

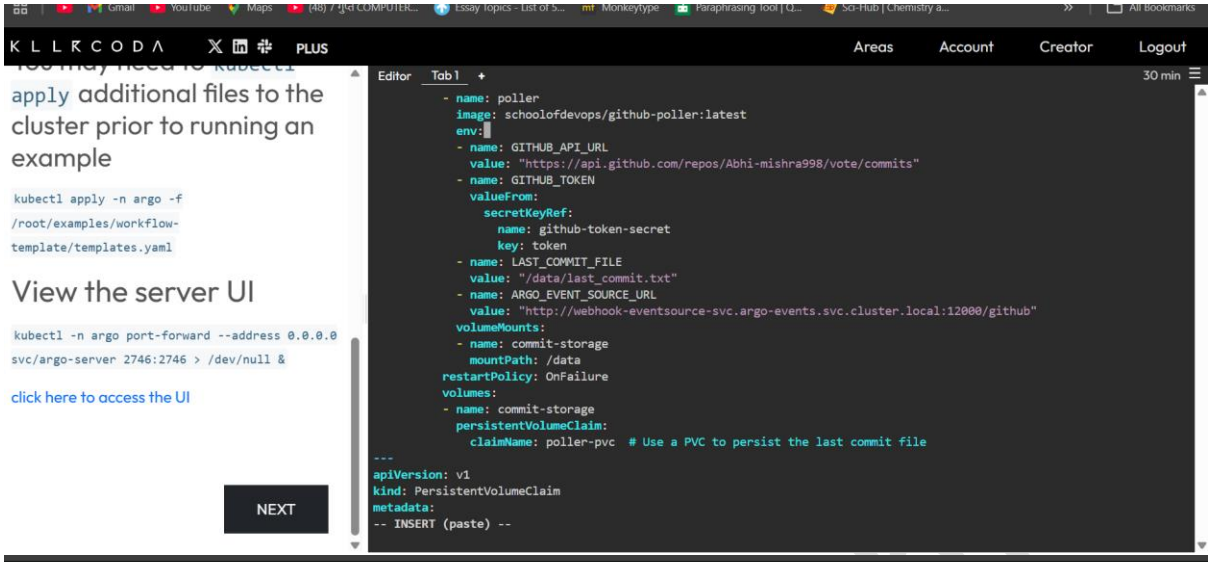

KLLRCODA
PLUS
Areas
Account
Creator
Logout
apply additional files to the cluster prior to running an example
kubectl apply -n argo -f /root/examples/workflow-template/templates.yaml
View the server UI
kubectl -n argo port-forward --address 0.0.0.0 svc/argo-server 2746:2746 > /dev/null &
click here to access the UI
NEXT
Editor
Tab 1
23 min
Every 2.0s: kubectl get pods
controlplane: Sat Oct 25 16:38:14 2025
NAME READY STATUS RESTARTS AGE
argo-server-db6795d8-ppl2c 1/1 Running 0 34m
github-polling-job-29356836-gjjgd 0/1 Completed 0 2m14s
github-polling-job-29356837-9zn54 0/1 Completed 0 74s
github-polling-job-29356838-kjkrt 0/1 Completed 0 14s
minio-55556775bb-skngm 1/1 Running 0 37m
minio-job-48pnp 0/1 Completed 0 37m
workflow-controller-5b6f48d789-mxmp7 1/1 Running 0 34m

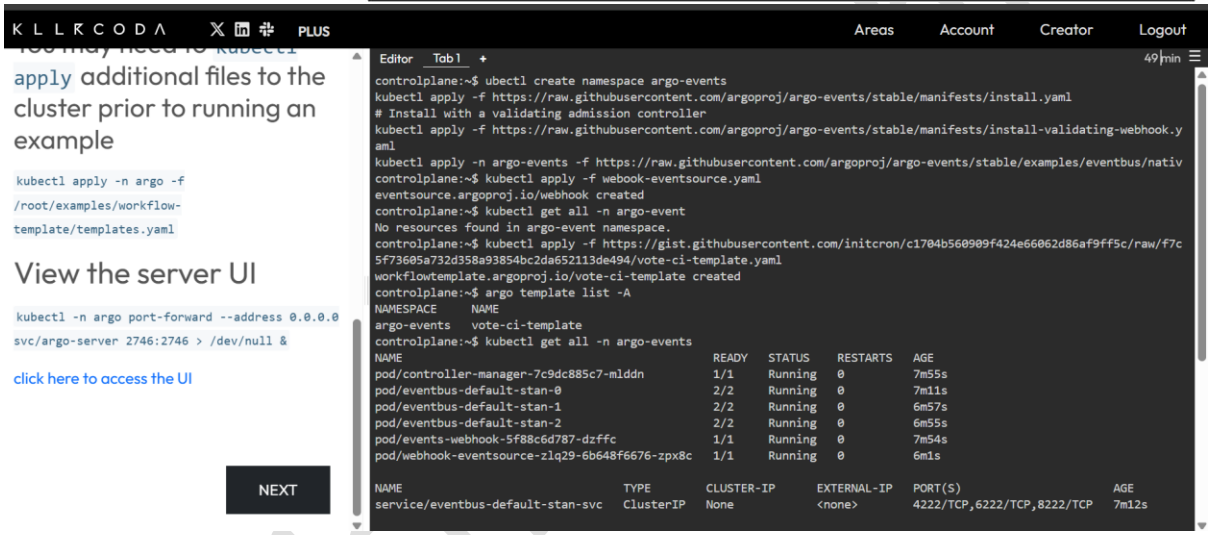

Gmail
YouTube
Maps
(48) 7 मुफ्त COMPUTER...
Essay Topics - List of 5...
Monkeytype
Paraphrasing Tool | Q...
Sci-Hub | Chemistry a...
All Bookmarks
KLLRCODA
PLUS
Areas
Account
Creator
Logout
apply additional files to the cluster prior to running an example
kubectl apply -n argo -f /root/examples/workflow-template/templates.yaml
View the server UI
kubectl -n argo port-forward --address 0.0.0.0 svc/argo-server 2746:2746 > /dev/null &
click here to access the UI
NEXT
Editor
Tab 1
24 min
Every 2.0s: kubectl get pods
controlplane: Sat Oct 25 16:37:28 2025
NAME READY STATUS RESTARTS AGE
argo-server-db6795d8-ppl2c 1/1 Running 0 34m
github-polling-job-29356834-c6t2d 0/1 Completed 0 3m28s
github-polling-job-29356835-rxmvq 0/1 Completed 0 2m28s
github-polling-job-29356836-gjjgd 0/1 ContainerCreating 0 88s
github-polling-job-29356837-9zn54 0/1 ContainerCreating 0 28s
minio-55556775bb-skngm 1/1 Running 0 36m
minio-job-48pnp 0/1 Completed 0 36m
workflow-controller-5b6f48d789-mxmp7 1/1 Running 0 34m

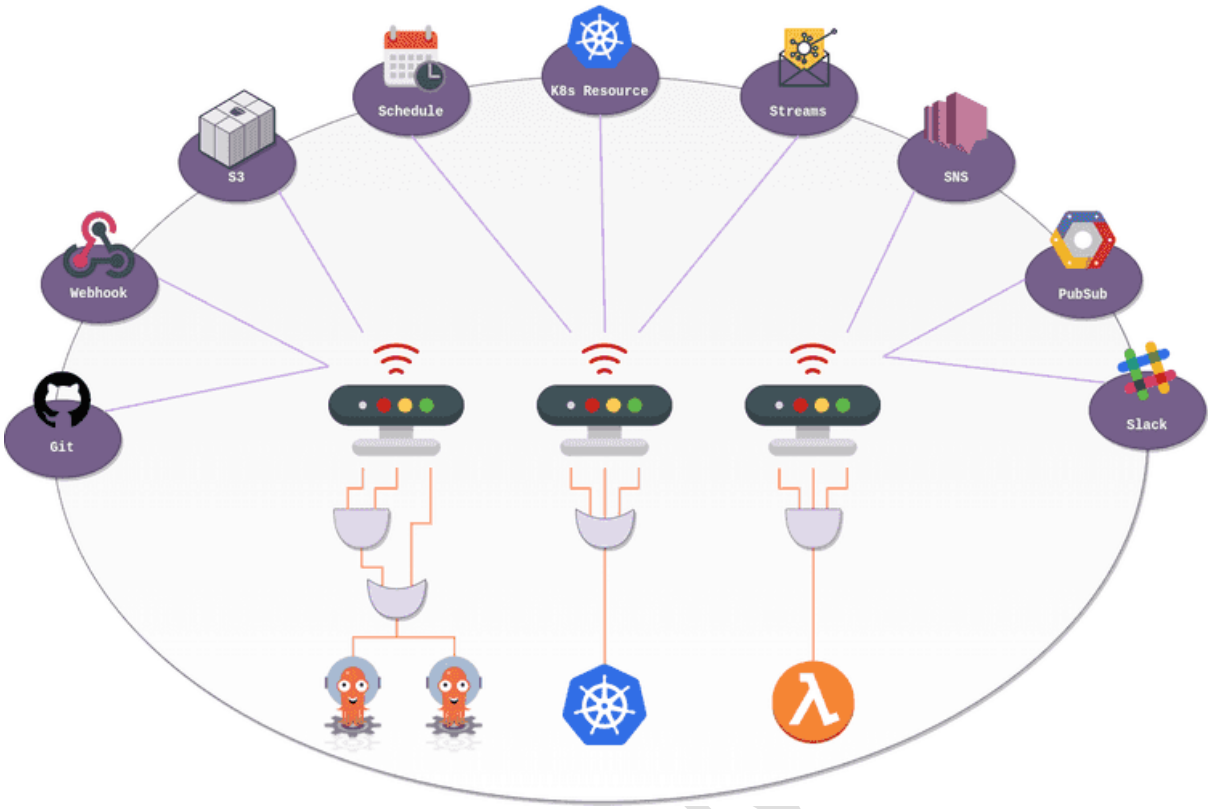

ad702953cc4a-10-244-4-43-2746.papa.r.killercoda.com/event-flow/argo-events
Event Flow
argo-events
EVENT FLOW
CREATE EVENT SOURCE
CREATE SENSOR
SHOW EVENT-FLOW
SHOW WORKFLOWS
COLLAPSE/EXPAND HIDDEN NODES
argo-events
Search
example
github
polling-sensor
launch-vote-ci
ci-pipeline-bxvbk
GET HELP


KILLERCODA
PLUS
Areas
Account
Creator
Logout
apply additional files to the cluster prior to running an example
kubectl apply -n argo -f /root/examples/workflow-template/templates.yaml
View the server UI
kubectl -n argo port-forward --address 0.0.0.0 svc/argo-server 2746:2746 > /dev/null &
click here to access the UI
NEXT
Editor
Tab 1
27 min
Every 2.0s: kubectl get pods
controlplane: Sat Oct 25 16:34:43 2025
NAME READY STATUS RESTARTS AGE
argo-server-db6795d8-ppl2c 1/1 Running 0 31m
github-polling-job-29356834-c6t2d 0/1 ContainerCreating 0 43s
minio-55556775bb-skngm 1/1 Running 0 33m
minio-job-48pnp 0/1 Completed 0 33m
workflow-controller-5b6f48d789-mxmp7 1/1 Running 0 31m


KILLERCODA
PLUS
Areas
Account
Creator
Logout
apply additional files to the cluster prior to running an example
kubectl apply -n argo -f /root/examples/workflow-template/templates.yaml
View the server UI
kubectl -n argo port-forward --address 0.0.0.0 svc/argo-server 2746:2746 > /dev/null &
click here to access the UI
NEXT
Editor
Tab 1
28 min
controlplane:~$ vi poller-cronjob.yaml
controlplane:~$ kubectl create secret generic github-token-secret --from-literal=token=ghp_N9JflJYOLp0YCebRRid76qFLFd41lW63Mtgk1
secret/github-token-secret created
controlplane:~$ kubectl apply -f poller-cronjob.yaml
cronjob.batch/github-polling-job created
persistentvolumeclaim/poller-pvc created
controlplane:~$


KLLRCODA
PLUS
Areas
Account
Creator
Logout
apply additional files to the cluster prior to running an example
kubectl apply -n argo -f /root/examples/workflow-template/templates.yaml
View the server UI
kubectl -n argo port-forward --address 0.0.0.0 svc/argo-server 2746:2746 > /dev/null &
click here to access the UI
NEXT
Editor
Tab 1
30 min
- name: poller
image: schoolofdevops/github-poller:latest
env:
- name: GITHUB_API_URL
value: "https://api.github.com/repos/Abhi-mishra998/vote/commits"
- name: GITHUB_TOKEN
valueFrom:
secretKeyRef:
name: github-token-secret
key: token
- name: LAST_COMMIT_FILE
value: "/data/last_commit.txt"
- name: ARGO_EVENT_SOURCE_URL
value: "http://webhook-eventsource-svc.argo-events.svc.cluster.local:12000/github"
volumeMounts:
- name: commit-storage
mountPath: /data
restartPolicy: OnFailure
volumes:
- name: commit-storage
persistentVolumeClaim:
claimName: poller-pvc # Use a PVC to persist the last commit file
---
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
-- INSERT (paste) --


KLLRCODA
PLUS
Areas
Account
Creator
Logout
apply additional files to the cluster prior to running an example
kubectl apply -n argo -f /root/examples/workflow-template/templates.yaml
View the server UI
kubectl -n argo port-forward --address 0.0.0.0 svc/argo-server 2746:2746 > /dev/null &
click here to access the UI
NEXT
Editor
Tab 1
49 min
controlplane:~$ ubectl create namespace argo-events
kubectl apply -f https://raw.githubusercontent.com/argoproj/argo-events/stable/manifests/install.yaml
Install with a validating admission controller
kubectl apply -f https://raw.githubusercontent.com/argoproj/argo-events/stable/manifests/install-validating-webhook.yaml
kubectl apply -n argo-events -f https://raw.githubusercontent.com/argoproj/argo-events/stable/examples/eventbus/nativ
controlplane:~$ kubectl apply -f webook-eventsource.yaml
eventsource.argoproj.io/webhook created
controlplane:~$ kubectl get all -n argo-event
No resources found in argo-event namespace.
controlplane:~$ kubectl apply -f https://gist.githubusercontent.com/initcron/c1704b560909f424e66062d86af9ff5c/raw/f7c5f73605a732d358a93854bc2da652113de494/vote-ci-template.yaml
workflowtemplate.argoproj.io/vote-ci-template created
controlplane:~$ argo template list -A
NAMESPACE NAME
argo-events vote-ci-template
controlplane:~$ kubectl get all -n argo-events
NAME READY STATUS RESTARTS AGE
pod/controller-manager-7c9dc885c7-mlddn 1/1 Running 0 7m55s
pod/eventbus-default-stan-0 2/2 Running 0 7m11s
pod/eventbus-default-stan-1 2/2 Running 0 6m57s
pod/eventbus-default-stan-2 2/2 Running 0 6m55s
pod/events-webhook-5f88c6d787-dzffc 1/1 Running 0 7m54s
pod/webhook-eventsource-zlq29-6b648f6676-zpx8c 1/1 Running 0 6m1s
NAME TYPE CLUSTER-IP EXTERNAL-IP PORT(S) AGE
service/eventbus-default-stan-svc ClusterIP None <none> 4222/TCP,6222/TCP,8222/TCP 7m12s



The Blue-Green Deployment: Why It Might Be Your Key to Flawless Software Updates

Let me paint you a picture.

Imagine you're a software engineer at a fast-growing e-commerce startup. It's Black Friday, the day when millions of people are hitting your website to shop for deals. The pressure is on. Your team has been working hard to roll out a new feature, but you're concerned that deploying it could disrupt the experience of thousands of users who are trying to make purchases.

You've heard about Kubernetes' rolling updates, the default strategy that replaces pods one by one with new versions. It sounds great on paper because it doesn't cause downtime, but what if something goes wrong during the update? What if an error in the new version causes customer checkout to fail, right in the middle of a record shopping day? Rolling back could be complicated and leave some customers using the old version while others struggle with the new, buggy release.

Enter **Blue-Green Deployment**.

This strategy can be your insurance against the chaos of failed deployments. And when combined with a tool like **Argo Rollouts**, which offers more advanced deployment strategies than Kubernetes' default, it takes things up a notch.

Let's dive in and see why Blue-Green Deployment, combined with Argo Rollouts, could be the hero of your next release.

What is Blue-Green Deployment?

At its core, Blue-Green Deployment is a deployment strategy that keeps two environments active at the same time: **Blue** and **Green**.

- The **Blue environment** is your existing, stable production environment.

- The **Green environment** is the new version of your application, with all your latest changes.

Instead of updating your application in-place like Kubernetes' rolling updates do, Blue-Green keeps your current version (Blue) running, while a complete copy of the new version (Green) is spun up in parallel. Once you're confident that Green is working correctly (you can test it in a staging-like environment), you **flip the switch** and route traffic to Green. Blue is still around, but out of the spotlight.

If things go wrong with the new version, you simply switch traffic back to Blue—no downtime, no complex rollbacks. It's like having an undo button for deployments.

Why Blue-Green Over Rolling Updates?

In Kubernetes, a **rolling update** gradually replaces old pods with new ones, ensuring there's no downtime. Sounds perfect, right?

Well, not always.

The problem with rolling updates is that you don't get an isolated environment for your new version. The update happens gradually, and there might be a period where some users are accessing the new version while others are still on the old version. If the new version has a critical bug, you're left trying to scramble a fix while traffic is hitting both versions. You could end up with inconsistent user experiences, and rollback can become messy.

With Blue-Green, you avoid this altogether because you have a clean separation. Only once you're 100% sure that the Green environment works flawlessly, you route all traffic to it at once. If something does go wrong, you just switch back to Blue without anyone noticing.

Now, let's make it even better with **Argo Rollouts**.

Why Use Argo Rollouts for Blue-Green Deployments?

Argo Rollouts is a Kubernetes controller that offers advanced deployment strategies like **Blue-Green**, **Canary**, and **Progressive Delivery**. While Kubernetes' default deployment controller can handle basic rolling updates, it lacks support for sophisticated strategies like Blue-Green or Canary out of the box.

Here's where Argo Rollouts shines: it enhances Kubernetes deployments, giving you more control, flexibility, and visibility into how your application is deployed. It's specifically designed to handle the types of challenges that arise in modern cloud-native applications.

Let's use an example to illustrate this.

An Example: Deploying a New Checkout Feature

Going back to our Black Friday scenario, imagine your team has developed a new checkout feature to improve the user experience. You're using Kubernetes, and you want to avoid the risks of rolling updates. Instead, you decide to use **Argo Rollouts** to manage a **Blue-Green Deployment**.

Here's how it might go:

1. **Set up the Blue environment**: Your current version of the checkout service is running, handling thousands of users. This is your Blue environment.

2. **Deploy the Green environment**: With Argo Rollouts, you create a new version of the checkout service—the Green environment—without interrupting the Blue environment. This includes spinning up new pods for the updated checkout service in parallel.
3. **Test the Green environment**: You run automated tests or manual QA on the Green environment, ensuring that everything works as expected.
4. **Traffic switch**: Once the Green environment is confirmed to be stable, Argo Rollouts allows you to easily route all traffic to the new version. Users are now hitting the Green environment.
5. **Monitoring and rollback**: If something unexpected happens after switching traffic to Green, Argo Rollouts gives you a rollback mechanism. With a simple command, you can redirect traffic back to the Blue environment, minimizing the impact on users.

Argo Rollouts also integrates with monitoring tools like Prometheus, so you can automatically trigger a rollback if performance metrics drop below a certain threshold.

Benefits of Argo Rollouts with Blue-Green

1. **No Downtime**: Blue-Green deployments ensure zero downtime. You can test your new version thoroughly without affecting users.
2. **Fast Rollback**: With Argo Rollouts, you can instantly switch back to the Blue version if any issues arise. It's seamless and quick, unlike the manual rollbacks you'd have to perform with rolling updates.
3. **Better User Experience**: Since all users are switched to the new version at once, there's no inconsistency where some users are on the old version and others are on the new. Everyone gets the same, predictable experience.
4. **Granular Control**: Argo Rollouts gives you more control over the deployment process. You can decide how and when traffic is shifted, and you can monitor the performance of the new version in real-time.
5. **Improved Automation**: Combined with tools like Prometheus for monitoring, Argo Rollouts can automate rollbacks if the new version doesn't meet performance standards. This reduces the risk of prolonged issues impacting users.

Conclusion

Blue-Green Deployment offers a simple yet powerful way to reduce risk and ensure a smooth user experience during software updates. By separating the old and new versions of your application, you gain the safety net of a fast rollback with zero downtime.

When paired with **Argo Rollouts**, Blue-Green Deployment becomes even more robust. You gain advanced control over the deployment process, real-time monitoring, and automated rollbacks, all while reducing the complexities of traditional Kubernetes deployments.

The Blue-Green Deployment: Why It Might Be Your Key to Flawless Software Updates

It’s a regular Tuesday morning at an IT operations center for a large multinational company. Suddenly, support tickets start pouring in—computers across the globe are freezing up. Employees are unable to access systems, deadlines are missed, and operations come to a grinding halt.

After a frantic investigation, the culprit is revealed: a bug in a newly released version of the **CrowdStrike endpoint protection software**. The software update had been pushed globally to Windows machines, and within minutes, an entire workforce was crippled. Millions of devices were impacted.

Could this massive, costly incident have been prevented? Yes. One strategy that could have made all the difference is **Progressive Canary Releases**.

Let’s dive into how **canary deployments** work and why they could save you from the nightmare of global outages.

What is a Progressive Canary Release?

A **Canary Release** is a deployment strategy where new features or changes are released to a small subset of users first, while the majority of users continue using the stable, older version of the software. This is named after the concept of “canaries in coal mines,” where miners would take canaries into the mines to detect dangerous gases. If the canary had a problem, it was an early warning sign to evacuate.

In the same way, a **Progressive Canary Release** incrementally shifts traffic from the old version to the new version in phases. Each phase targets a progressively larger percentage of users. You monitor the behavior of the canary group at each stage, watching for errors, performance degradation, or unexpected behavior. Only when you’re confident that the new release is stable and performing well, do you continue rolling it out to the next group of users.

In short, it's a way to **minimize risk** by catching potential issues early, on a smaller scale, before they affect everyone.

The Global Windows Outage: What Happened?

Let’s go back to that Tuesday morning with the **CrowdStrike software**. A critical bug in a new version of the software caused Windows systems to freeze, impacting millions of devices worldwide. Here’s a high-level view of what went wrong:

1. **Global Rollout**: The software update was pushed to a massive number of devices simultaneously.
2. **No Early Warning**: Without a smaller test group to detect the issue, the bug went live for all users at once.
3. **Difficult Rollback**: Rolling back a global update, especially when machines are non-functional, is a massive logistical challenge.

Now, imagine the team had used **Progressive Canary Releases**. Here’s how things might have gone differently.

How Canary Releases Could Have Prevented the Outage

With a **Progressive Canary Release**, the update would have been rolled out in controlled stages. Let's break down what that would look like.

Step 1: Initial Canary Release to a Small Group

Instead of deploying the new version of CrowdStrike to millions of devices at once, it's first deployed to just **1% of users**. This canary group might consist of a small subset of the company's devices across different geographies, operating system versions, and use cases.

```
+--------------------------------+
|          Users                 |
+--------------------------------+
            |
            V
+--------------------+
|   Load Balancer    |
+--------------------+
            |
            V
+--------------------+
|    Canary (v2)     | <------ 1% of users only
|   New Version      |
+--------------------+
            |
            V
+--------------------+
|   Stable (v1)      | <------ 99% of users
| Current Version    |
+--------------------+
```

Monitoring: At this point, the operations team monitors the behavior of the canary group closely. Are Windows systems freezing up? Is there an increase in resource usage or system crashes?

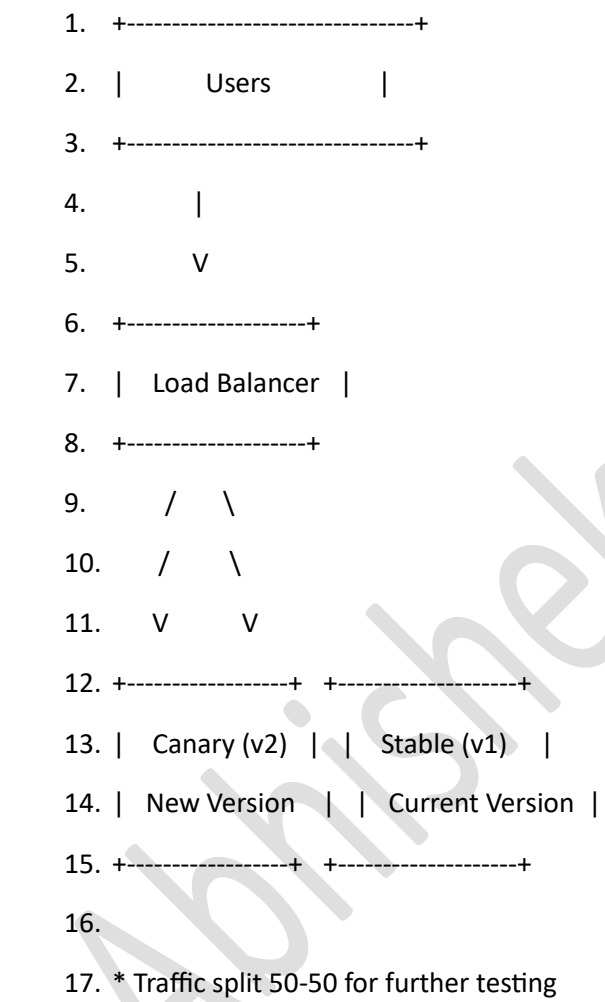
If the canary group reports problems—let's say, the same issue that caused the Windows freeze—developers can immediately halt the release.

Step 2: Gradual Traffic Shifting

If the initial canary deployment is successful (no bugs, no performance issues), traffic is **gradually shifted** to the new version in larger increments. For example:

- Phase 1: 1% of users get the new version.
- Phase 2: 10% of users are shifted to the new version.
- Phase 3: 25% of users are now on the new version.
- Phase 4: 50% of users receive the new version.

Each phase includes a period of **intensive monitoring**. The operations team checks error logs, performance metrics, and feedback from users. At any sign of trouble, the rollout can be paused or even reversed, keeping the majority of users safe.



Key Benefits:

- **Risk Containment**: If there is a bug (like the Windows freeze), it impacts a small number of users, not the entire global fleet.
- **Easy Rollback**: If problems arise at any stage, the system can quickly revert traffic back to the stable version (v1) without downtime.

Step 3: Full Rollout

Once the new version (v2) has been thoroughly tested in increments and no issues have been detected, the final stage is to route **100% of traffic** to the new version. At this point, the update has

been verified through multiple stages, and the risk of encountering major issues is dramatically reduced.

```
+---------------------------------+
|          Users                  |
+---------------------------------+
            |
            V
+--------------------+
|   Load Balancer    |
+--------------------+
            |
            V
+--------------------+
|    New Version     | <------ 100% of users
|     (v2)           |
+--------------------+

* New version is fully deployed across all users.
```

Why Canary Releases Work: Benefits of the Approach

1. **Early Detection of Bugs**: Canary releases catch issues early in the deployment process. In the case of the CrowdStrike outage, the Windows freeze bug would have been identified within the initial 1% of devices, long before it spread globally.
2. **Minimal User Impact**: Only a small fraction of users are impacted during the early stages. If an issue arises, you can roll back without causing widespread disruption.
3. **Better Monitoring & Feedback Loops**: Canary releases give you the opportunity to gather feedback from real-world users while still controlling risk. This allows teams to course-correct before proceeding with a wider rollout.
4. **Controlled Rollback**: If something does go wrong, the rollback process is simple and contained. You can return to the previous stable version for the small percentage of impacted users.
5. **Safer Deployments in Complex Environments**: In environments where you have many different operating systems, hardware configurations, or global users, canary deployments allow for testing across different environments before full rollout.

How Argo Rollouts Can Supercharge Canary Releases

While Kubernetes provides basic support for canary deployments, **Argo Rollouts** adds powerful capabilities for **Progressive Delivery**. Here's how:

- **Automated Traffic Shifting**: Argo Rollouts can automate the process of shifting traffic between versions, using predefined rules for how much traffic should be routed to the new version at each stage.
- **Metrics-Driven Rollouts**: You can integrate with tools like Prometheus to monitor key metrics during each phase. If errors, latency, or CPU usage exceed thresholds, Argo Rollouts can pause or even automatically rollback the deployment.
- **Detailed Progress Visibility**: Argo Rollouts offers a dashboard where you can track the status of each phase in real-time, making it easier for your team to make informed decisions.

Conclusion

A **Progressive Canary Release** could have saved the day during the CrowdStrike Windows outage. By deploying the new version incrementally and monitoring its behavior at each stage, the critical bug would have been detected in the early stages, preventing a global crisis.

When combined with tools like **Argo Rollouts**, canary deployments offer even greater control and automation, allowing teams to deliver software updates with confidence, reduced risk, and a clear rollback plan. This strategy not only minimizes user impact but also ensures your systems remain stable and resilient, even in complex global environments.