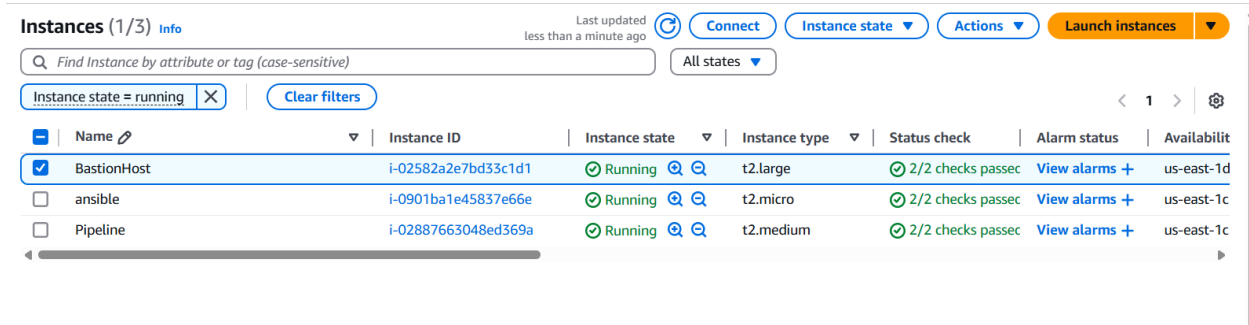


## Assessment Overview

Created Three Servers: Ansible Master, Bastion Host (Slave) and Pipeline (Jenkins) (Slave).



The screenshot shows the AWS Management Console 'Instances' page. It displays three EC2 instances that are all in a 'Running' state. The instances are named 'BastionHost', 'ansible', and 'Pipeline'. Each instance has a status check of '2/2 checks passed' and is located in the 'us-east-1c' availability zone. The 'BastionHost' is a 't2.large' instance, while 'ansible' and 'Pipeline' are 't2.micro' and 't2.medium' instances respectively.

	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability
<input checked="" type="checkbox"/>	BastionHost	i-02582a2e7bd33c1d1	Running	t2.large	2/2 checks passed	View alarms +	us-east-1c
<input type="checkbox"/>	ansible	i-0901ba1e45837e66e	Running	t2.micro	2/2 checks passed	View alarms +	us-east-1c
<input type="checkbox"/>	Pipeline	i-02887663048ed369a	Running	t2.medium	2/2 checks passed	View alarms +	us-east-1c

All the installation is done through Ansible.

1. Created a Bastion Host with required configurations installed using Ansible making it only source of contact with Kubernetes Cluster present in Private Subnet.

```
ubuntu@ip-172-31-46-31:~$ kubectl version --client
Client Version: v1.31.0
Kustomize Version: v5.4.2
ubuntu@ip-172-31-46-31:~$
```

```
[root@ip-172-31-19-160 bastion-playbooks]# vi main.yml
[root@ip-172-31-19-160 bastion-playbooks]# ansible-playbook -i inventory.ini main.yml
```

```
PLAY [bastionhost] *****

TASK [Gathering Facts] *****
ok: [ubuntu@54.172.91.255]

TASK [eksctl : Copy bash script to server] *****
changed: [ubuntu@54.172.91.255]

TASK [eksctl : execute bash script on server] *****
changed: [ubuntu@54.172.91.255]

TASK [prerequisites : install basic packages] *****
ok: [ubuntu@54.172.91.255]

TASK [prerequisites : change permission for /opt] *****
ok: [ubuntu@54.172.91.255]

TASK [kubectl : Copy bash script to server] *****
ok: [ubuntu@54.172.91.255]

TASK [kubectl : execute bash script on server] *****
changed: [ubuntu@54.172.91.255]

TASK [kubectl : Installing Skaffold] *****
changed: [ubuntu@54.172.91.255]

TASK [helm : install helm from direct url] *****
changed: [ubuntu@54.172.91.255]
```

## Created Kubernetes Cluster with nginx Controller

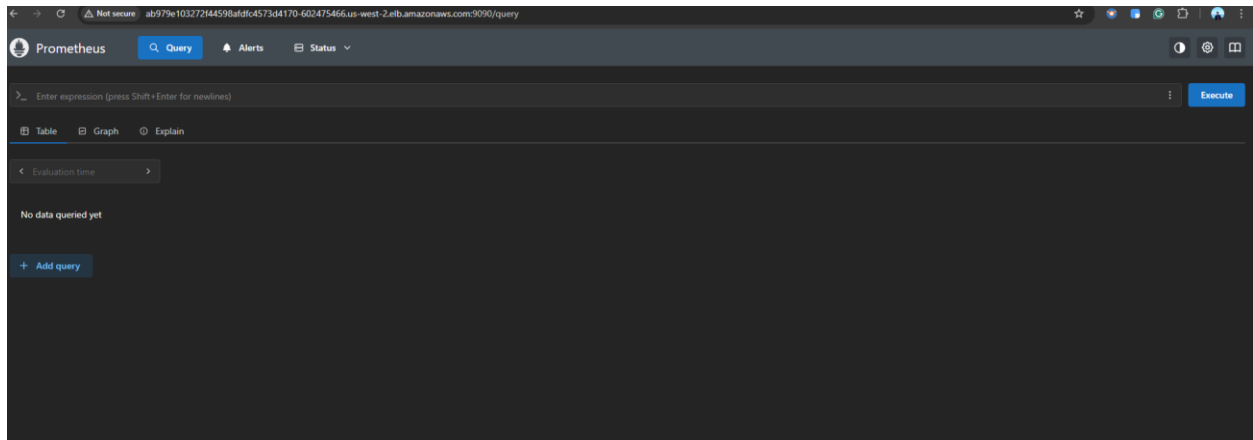
```
root@ip-172-31-46-3:/home/ubuntu# eksctl create cluster --name three-tier-cluster --region us-west-2 --node-type t2.medium --nodes-min 2 --nodes-max 2
2025-05-08 13:59:36 [i] eksctl version 0.207.0
2025-05-08 13:59:36 [i] using region us-west-2
2025-05-08 13:59:36 [i] skipping us-west-2d from selection because it doesn't support the following instance type(s): t2.medium
2025-05-08 13:59:36 [i] setting availability zones to [us-west-2b us-west-2c us-west-2a]
2025-05-08 13:59:36 [i] subnets for us-west-2b - public:192.168.0.0/19 private:192.168.86.0/19
2025-05-08 13:59:36 [i] subnets for us-west-2c - public:192.168.32.0/19 private:192.168.128.0/19
2025-05-08 13:59:36 [i] subnets for us-west-2a - public:192.168.64.0/19 private:192.168.160.0/19
2025-05-08 13:59:36 [i] nodegroup "ng-029198dc" will use "" [AmazonLinux2/1.32]
2025-05-08 13:59:36 [i] using Kubernetes version 1.32
2025-05-08 13:59:36 [i] creating EKS cluster "three-tier-cluster" in "us-west-2" region with managed nodes
2025-05-08 13:59:36 [i] will create 2 separate CloudFormation stacks for cluster itself and the initial managed nodegroup
2025-05-08 13:59:36 [i] if you encounter any issues, check CloudFormation console or try 'eksctl utils describe-stacks --region=us-west-2 --cluster=three-tier-cluster'
2025-05-08 13:59:36 [i] Kubernetes API endpoint access will use default of [publicAccess=true, privateAccess=false] for cluster "three-tier-cluster" in "us-west-2"
2025-05-08 13:59:36 [i] CloudWatch logging will not be enabled for cluster "three-tier-cluster" in "us-west-2"
2025-05-08 13:59:36 [i] you can enable it with 'eksctl utils update-cluster-logging --enable-types=(SPECIFY-YOUR-LOG-TYPES-HERE (e.g. all)) --region=us-west-2 --cluster=three-tier-cluster'
2025-05-08 13:59:36 [i] default addons vpc-cni, kube-proxy, coredns, metrics-server were not specified, will install them as EKS addons
2025-05-08 13:59:36 [i]
2 sequential tasks: [ create cluster control plane "three-tier-cluster",
  2 sequential sub-tasks: [
    2 sequential sub-tasks: [
      1 task: [ create addons ],
      wait for control plane to become ready,
    ],
    create managed nodegroup "ng-029198dc",
  ]
]
2025-05-08 13:59:36 [i] building cluster stack "eksctl-three-tier-cluster-cluster"
2025-05-08 13:59:37 [i] deploying stack "eksctl-three-tier-cluster-cluster"
2025-05-08 14:00:07 [i] waiting for CloudFormation stack "eksctl-three-tier-cluster-cluster"
2025-05-08 14:00:37 [i] waiting for CloudFormation stack "eksctl-three-tier-cluster-cluster"
2025-05-08 14:01:37 [i] waiting for CloudFormation stack "eksctl-three-tier-cluster-cluster"
^C

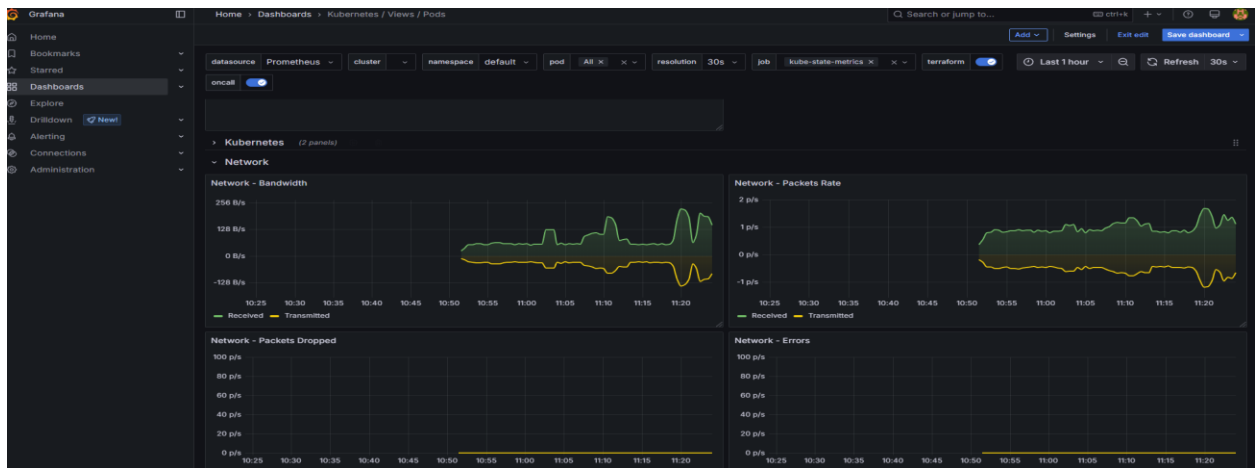
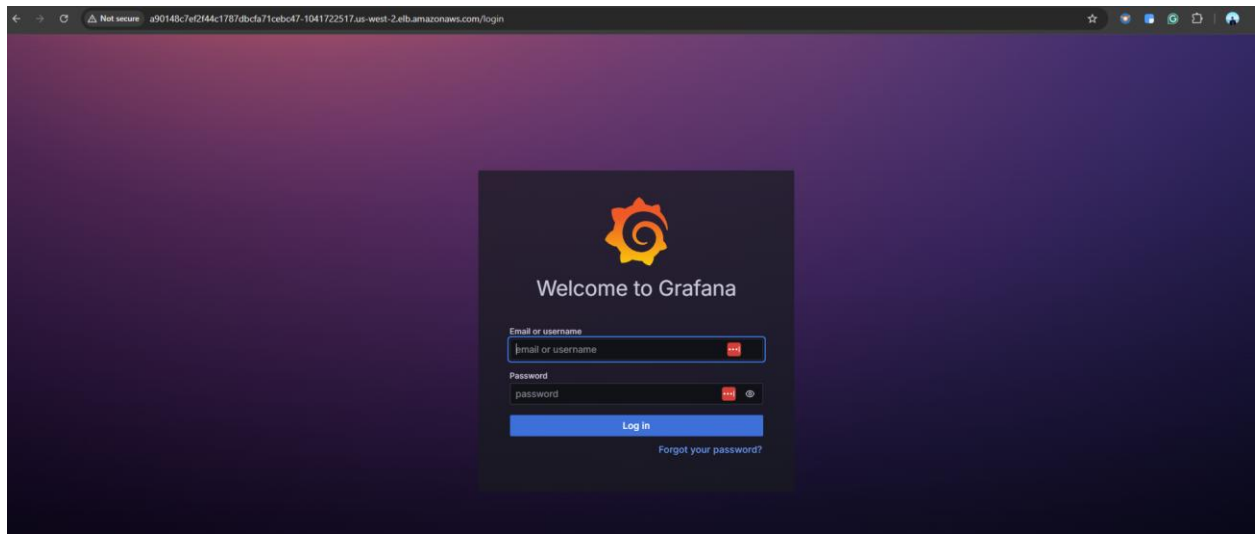
root@ip-172-31-46-3:/home/ubuntu/Build-A-High-Availability-System-Design-Assessment/K8s Manifest - Hot# kubectl get pods -n ingress-nginx
NAME                                READY   STATUS    RESTARTS   AGE
ingress-nginx-controller-6885cfc548-pwnwp  1/1     Running   0          27s
root@ip-172-31-46-3:/home/ubuntu/Build-A-High-Availability-System-Design-Assessment/K8s Manifest - Hot# kubectl get ingress
NAME                                CLASS    HOSTS                                ADDRESS                                                                 PORTS   AGE
backend-ingress                    nginx    *                                     a109c25fa162c4bb8ad2750a1049639a-1514297137.us-west-2.elb.amazonaws.com  80      36m
```

```
← → ↻ ⚠ Not secure a92cc198e7ec24fe4b79ae377a85e264-1947725967.us-west-2.elb.amazonaws.com/hello ☆
pretty-print [x]
"message": "Hello from Kubernetes REST API!"
```

## Prometheus and Grafana Setup using Helm:

```
meta.helm.sh/release-namespace: prometheus
creationTimestamp: "2025-05-08T14:50:36Z"
labels:
  app: kube-prometheus-stack-prometheus
  app.kubernetes.io/instance: stable
  app.kubernetes.io/managed-by: Helm
  app.kubernetes.io/part-of: kube-prometheus-stack
  app.kubernetes.io/version: 72.1.1
  chart: kube-prometheus-stack-72.1.1
  heritage: Helm
  release: stable
  self-monitor: "true"
name: stable-kube-prometheus-sta-prometheus
namespace: prometheus
resourceVersion: "8395"
uid: b979e103-272f-4459-8afd-fc4573d4170c
pec:
  clusterIP: 10.100.238.69
  clusterIPs:
    - 10.100.238.69
  internalTrafficPolicy: Cluster
  ipFamilies:
    - IPv4
  ipFamilyPolicy: SingleStack
  ports:
    - name: http-web
      port: 9090
      protocol: TCP
      targetPort: 9090
    - appProtocol: http
      name: reloader-web
      port: 8080
      protocol: TCP
      targetPort: reloader-web
  selector:
    app.kubernetes.io/name: prometheus
    operator.prometheus.io/name: stable-kube-prometheus-sta-prometheus
  sessionAffinity: None
  type: LoadBalancer[]
status:
  loadBalancer: {}
```





## Installing Jenkins and SonarQube for pipeline:

```
TASK [Gathering Facts] *****
[WARNING]: Platform linux on host ec2-user@54.224.69.237 is using the discovered Python interpreter at /usr/bin/python3.9, but future installation of another Python interpreter could change the meaning of the path. See https://docs.ansible.com/ansible-core/2.15/reference_appendices/interpreter_discovery.html for more information.
ok: [ec2-user@54.224.69.237]

TASK [jenkins : Ensure Java is installed (Amazon Linux 2)] *****
ok: [ec2-user@54.224.69.237]

TASK [jenkins : Add Jenkins repo] *****
ok: [ec2-user@54.224.69.237]

TASK [jenkins : Import Jenkins GPG key] *****
ok: [ec2-user@54.224.69.237]

TASK [jenkins : Install Jenkins] *****
ok: [ec2-user@54.224.69.237]

TASK [jenkins : Enable and start Jenkins service] *****
ok: [ec2-user@54.224.69.237]

TASK [SonarQube : Install Docker] *****
changed: [ec2-user@54.224.69.237]

TASK [SonarQube : Ensure Docker service is running] *****
changed: [ec2-user@54.224.69.237]

TASK [SonarQube : Add ec2-user to docker group] *****
changed: [ec2-user@54.224.69.237]

TASK [SonarQube : Pull SonarQube Docker image] *****
changed: [ec2-user@54.224.69.237]

TASK [SonarQube : Run SonarQube container] *****
changed: [ec2-user@54.224.69.237]

PLAY RECAP *****
ec2-user@54.224.69.237 : ok=11 changed=5 unreachable=0 failed=0 skipped=0 rescued=0 ignored=0
```

Getting Started

# Getting Started

✓ Folders	✓ OWASP Markup Formatter	✓ Build Timeout	✓ Credentials Binding	** Ionicons API
✓ Timestamp	○ Workspace Cleanup	○ Ant	○ Gradle	** OWASP Markup Formatter
○ Pipeline	○ GitHub Branch Source	○ Pipeline: GitHub Groovy Libraries	○ Pipeline Graph View	** ASH API
○ Git	○ SSH Build Agents	○ Matrix Authorization Strategy	○ RAM Authentication	** JSCN Path API
○ LDAP	○ Email Extension	○ Mailer	○ Dark Theme	** Struts
				** Pipeline: Step API
				** Token Macro
				** Build Timeout
				** SonarQube API
				** Credentials
				** Plain Credentials
				** Variant
				** SSH Credentials
				** Credentials Binding
				** SCH API
				** Pipeline: API
				** commons-lang3 v3.x Jenkins API
				** Timestamp
				** Carrefour API
				** Script Security
				** JavaBeans Activation Framework (JAF) API
				** JAXB
				** SnakeYAML API
				** JSCN API
				** Jackson 2 API
				** commons-text API
				** Pipeline: Supporting APIs
				** Plugin: Utilities API
				** Font Awesome API
				** Bootstrap 5 API
				** JQuery API
				** - required dependency

Jenkins 2.504.1

⚠ Embedded database should be used for evaluation purposes only. It doesn't support scaling, upgrading to a new SonarQube Server version, or migration to another database engine. [Learn more](#)

☆ Node / main ✓ ?

Overview Issues Security Hotspots Measures Code Activity

Project Settings Project Information



Quality Gate ⓘ

**Passed**

Last analysis 51 seconds ago

⚠ The last analysis has warnings. [See details](#)

New Code

Overall Code

Security

0 Open issues

A

Reliability

0 Open issues

A

Maintainability

0 Open issues

A

Accepted issues

0

Valid issues that were not fixed

ⓘ

Coverage

0.0%

On 7 lines to cover.

Duplications

0.0%

On 12 lines.