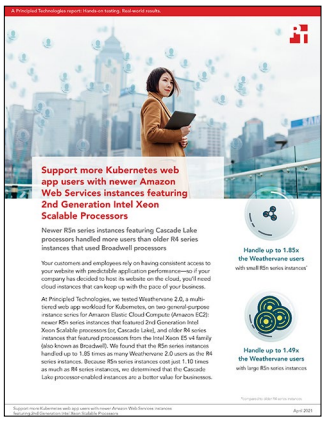

The science behind the report:

Support more Kubernetes web app users with newer Amazon Web Services instances featuring 2nd Generation Intel Xeon Scalable Processors

This document describes what we tested, how we tested, and what we found. To learn how these facts translate into real-world benefits, read the report Support more Kubernetes web app users with newer Amazon Web Services instances featuring 2nd Generation Intel Xeon Scalable Processors.

We concluded our hands-on testing on March 26, 2021. During testing, we determined the appropriate hardware and software configurations and applied updates as they became available. The results in this report reflect configurations that we finalized on March 25, 2021 or earlier. Unavoidably, these configurations may not represent the latest versions available when this report appears.

Our results

To learn more about how we have calculated the wins in this report, go to http://facts.pt/calculating-and-highlighting-wins. Unless we state otherwise, we have followed the rules and principles we outline in that document.

Table 1: Results of Weathervane 2.0 testing on R5n and R4 series instances for Amazon Elastic Cloud Compute (Amazon EC2).

Instance name	Weathervane users
8vCPU instances	
r5n.2xlarge	30,200
r4.2xlarge	16,300
32vCPU results	
r5n.8xlarge	82,500
r4.8xlarge	55,150

System configuration information

Table 2: Detailed information on the Amazon EC2 R5n series instances we tested.

Cluster configuration information	3x 8vCPU Cascade Lake VM	3x 32vCPU Cascade Lake VM
Tested by	Principled Technologies	Principled Technologies
Test date	03/16/2021	03/18/2021
CSP / Region	Amazon Web Services US East 1	Amazon Web Services US East 1
Workload & version	Weathervane 2.0.9	Weathervane 2.0.9
WL specific parameters	3x small2 app instances, useAppServerLimits : false	8x small2 app instances, useAppServerLimits : false
Iterations and result choice	3 runs, median	3 runs, median
Server platform	3x r5n.2xlarge	3x r5n.8xlarge
BIOS name and version	Amazon EC2 1.0, 10/16/2017	Amazon EC2 1.0, 10/16/2017
Operating system name and version/build number	Amazon Linux 2	Amazon Linux 2
Date of last OS updates/ patches applied	03/16/2021	03/18/2021
Processor		
Number of processors	1	1
Vendor and model	Intel®Xeon® Platinum 8259CL	Intel Xeon Platinum 8259CL
Core count (per processor)	24	24
Core frequency (GHz)	2.50	2.50
Stepping	7	7
Hyper-Threading	Yes	Yes
Turbo	Yes	Yes
Number of vCPU per VM	8	32
Memory module(s)		
Total memory in system (GiB)	64	256
NVMe memory present?	No	No
Total memory (DDR+NVMe RAM)	64	256
General hardware		
Storage: NW or Direct Att / Instance	NW Att	NW Att
Network BW / Instance	N/A	N/A
Storage BW / Instance	N/A	N/A

Cluster configuration information	3x 8vCPU Cascade Lake VM	3x 32vCPU Cascade Lake VM
Local storage		
OS		
Number of drives	1	1
Drive size (GB)	100	100
Drive information (speed, interface, type)	Standard SSD	Standard SSD
Data drive		
Number of drives	12	32
Drive size (GB)	3x 60GB, 3x 10GB, 6x 5GB	8x 60GB, 8x 10GB, 16x 5GB
Drive information (speed, interface, type)	Standard SSD	Standard SSD
Network adapter		
Vendor and model	Amazon ENA	Amazon ENA
Number and type of ports	1x 25Gb	1x 25Gb

Table 3: Detailed information on the Amazon EC2 R4 series instances we tested.

Cluster configuration information	3x 8vCPU Broadwell VM	3x 32vCPU Broadwell VM
Tested by	Principled Technologies	Principled Technologies
Test date	03/23/2021	03/25/2021
CSP / Region	Amazon Web Services US East 1	Amazon Web Services US East 1
Workload & version	Weathervane 2.0.9	Weathervane 2.0.9
WL specific parameters	3x small2 app instances, useAppServerLimits : false	8x small2 app instances, useAppServerLimits : false
Iterations and result choice	3 runs, median	3 runs, median
Server platform	3x r4.2xlarge	3x r4.8xlarge
BIOS name and version	Xen 4.2.amazon 8/24/2006	Xen 4.2.amazon 8/24/2006
Operating system name and version/build number	Amazon Linux 2	Amazon Linux 2
Date of last OS updates/ patches applied	03/23/2021	03/25/2021
Processor		
Number of processors	1	1
Vendor and model	Intel Xeon E5-2686 v4	Intel Xeon E5-2686 v4
Core count (per processor)	18	18
Core frequency (GHz)	2.30	2.30
Stepping	1	1
Hyper-Threading	Yes	Yes
Turbo	Yes	Yes
Number of vCPU per VM	8	32

Cluster configuration information	3x 8vCPU Broadwell VM	3x 32vCPU Broadwell VM
Memory module(s)		
Total memory in system (GiB)	61	244
NVMe memory present?	No	No
Total memory (DDR+NVMe RAM)	64	256
General hardware		
Storage: NW or Direct Att / Instance	NW Att	NW Att
Network BW / Instance	N/A	N/A
Storage BW / Instance	N/A	N/A
Local storage		
OS		
Number of drives	1	1
Drive size (GB)	100	100
Drive information (speed, interface, type)	Standard SSD	Standard SSD
Data drive		
Number of drives	12	32
Drive size (GB)	3x 60GB, 3x 10GB, 6x 5GB	8x 60GB, 8x 10GB, 16x 5GB
Drive information (speed, interface, type)	Standard SSD	Standard SSD
Network adapter		
Vendor and model	Amazon ENA	Amazon ENA
Number and type of ports	1x 10Gb	1x 10Gb

How we tested

Testing overview

For this project, we will test Amazon Elastic Kubernetes Service (EKS) cluster featuring older Intel processors vs. Cascade Lake versions. We will run the VMWare Weathervane benchmark on the AWS instances to show the increased cluster capabilities that customers can expect to see using the newer instance series vs. the older.

Creating the run-harness VM

This section contains the steps we took to create our run-harness VM.

Creating the run-harness VM

1. Log into the AWS console, and click EC2.
2. Click Instances, and click Launch instances.
3. Type `Ubuntu` into the search bar. On the Ubuntu 20.04 AMI, click Select.
4. For the instance type, select t2.large, and click Next: Configure Instance Details.
5. Click Next: Add Storage.
6. Click Next: Add Tags.
7. Click Next: Configure Security Group.
8. Configure rules to open the following ports:
    - 6443
    - 2379-2380
    - 22
    - 10255
    - 10250-10252
9. Click Review and Launch.
10. Click Launch.

Configuring the run-harness VM

1. Log into the run-harness VM using the ssh command in the Connect tab for the VM.
2. Install the latest updates on the VM:

```
sudo apt update
sudo apt-get upgrade -y
sudo reboot
```

3. Install unzip:

```
sudo apt-get unzip
```

4. Set up the prerequisites for Docker repository access:

```
sudo apt-get install \
apt-transport-https \
ca-certificates \
curl \
gnupg
```

5. Add the official GPG key for Docker:

```
curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo gpg --dearmor -o /usr/share/keyrings/
    docker-archive-keyring.gpg
```

6. Set up the stable Docker repository:

```
echo \
"deb [arch=amd64 signed-by=/usr/share/keyrings/docker-archive-keyring.gpg] https://download.docker.
    com/linux/ubuntu\
$(lsb_release -cd) stable" | sudo tee /etc/apt/sources.list.d/docker.list > /dev/null
```

7. Install Docker Engine:

```
sudo apt-get update
sudo apt-get install docker-ce docker-ce-cli containerd.io
```

8. Install git, if not already installed:

```
sudo apt-get install git
```

9. Clone the Weathervane repository to the run-harness VM:

```
git clone https://github.com/vmware/weathervane
```

10. Create the Weathervane Docker images (Note: If you do not have a Docker Hub account, you will need to create one to host the Docker images for the benchmark):

```
cd weathervane
./buildDockerImages.pl –username [Docker Hub username]
```

11. When prompted, input your Docker Hub password.
12. Once the build has finished, create a copy of the Weathervane configuration file:

```
cp weathervane.config.k8s.micro weathervane.config.k8s.quickstart
```

13. In the next section, we include the steps for editing the configuration file.

Creating and accessing the Elastic Kubernetes Service (EKS) cluster

This section contains the steps we took to create our Kubernetes cluster.

Creating the Kubernetes cluster

1. Using the steps from the previous section, create a second VM using the Centos 7 AMI and the t2.medium instance type.
2. Log into the EKS VM using the ssh command in the Connect tab.
3. Install latest updates on the VM:

```
sudo yum update
sudo yum upgrade -y
sudo reboot
```

4. Install the Amazon AWS CLI v2:

```
curl "https://awscli.amazonaws.com/awscli-exe-linux-x86_64.zip" -o "awscliv2.zip"
sudo yum -y install unzip
unzip awscliv2.zip
sudo /aws/install
aws configure
```

5. Enter the username, password, and AWS region to configure the AWS CLI credentials.
6. Install `kubectl`:

```
curl -o kubectl https://amazon-eks.s3.us-west-2.amazonaws.com/1.18.9/2020-11-02/bin/
    linux/amd64/kubectl
chmod +x /kubectl
```

7. Create an AWS VPC:

```
aws cloudformation create-stack --stack-name eks-stack --template-url https://s3.us-west-2.amazonaws.com/
amazon-eks/cloudformation/2020-10-29/amazon-eks-vpc-private-subnets.yaml
```

8. Create a file called `cluster-role-trust-policy.json` and edit it to the following:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "eks.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

9. Create the role:

```
aws iam create-role --role-name EKSClusterRole --assume-role-policy-document file://"cluster-role-trust-policy.json"
```

10. Attach the policy to the role:

```
aws iam attach-role-policy --policy-arn arn:aws:iam::aws:policy/AmazonEKSClusterPolicy --role-name EKSClusterRole
```

11. Create a key pair:

```
aws ec2 create-key-pair --region us-east-1 --key-name eks-public-key
```

12. Create the EKS cluster:

```
eksctl create cluster --name=eks-8cpu --ssh-access --ssh-public-key=eks-public-key --region=us-east-1 --without-nodegroup
```

13. Create a file called `nodegroups.yaml` and edit it to the following, changing the SUT instance type as needed:

```
apiVersion: eksctl.io/v1alpha5
kind: ClusterConfig

metadata:
  name: eks-8cpu
  region: us-east-1

nodeGroups:
  - name: ng-1-sut
    labels: { role: sut }
    instanceType: r5n.8xlarge
    desiredCapacity: 3
    ssh:
      enableSsm: true
    volumeSize: 100
  - name: ng-2-driver
    labels: { role: driver }
    instanceType: m5n.8xlarge
    desiredCapacity: 1
    ssh:
      enableSsm: true
volumeSize: 100
```

14. Create the nodegroups:

```
eksctl create nodegroup --config-file=nodegroup.yaml
```

15. Create a file called `gp2-immediate.yaml` and edit it to the following:

```
allowVolumeExpansion: true
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  annotations:
    kubectl.kubernetes.io/last-applied-configuration: |
      {"apiVersion":"storage.k8s.io/
    v1","kind":"StorageClass","metadata":{"annotations":{"storageclass.kubernetes.io/is-default-
    class":"true"},"name":"gp2-immediate"},"parameters":{"fsType":"ext4","type":"gp2"},"provisioner":
    "kubernetes.io/aws-ebs","volumeBindingMode":"Immediate"}
    storageclass.kubernetes.io/is-default-class: "true"
  creationTimestamp: "2021-02-17T16:36:53Z"
  managedFields:
  - apiVersion: storage.k8s.io/v1
    fieldsType: FieldsV1
    fieldsV1:
      f:allowVolumeExpansion: {}
      f:metadata:
        f:annotations:
          .: {}
          f:kubectl.kubernetes.io/last-applied-configuration: {}
          f:storageclass.kubernetes.io/is-default-class: {}
      f:parameters:
        .: {}
        f:fsType: {}
        f:type: {}
      f:provisioner: {}
      f:reclaimPolicy: {}
      f:volumeBindingMode: {}
    manager: kubectl
    operation: Update
    time: "2021-02-17T19:11:07Z"
  name: gp2-immediate
  resourceVersion: "27426"
  selfLink: /apis/storage.k8s.io/v1/storageclasses/gp2
  uid: 8d9c4a96-ed5d-4cef-9027-64a58b0d94f1
parameters:
  fsType: ext4
  type: gp2
provisioner: kubernetes.io/aws-ebs
reclaimPolicy: Delete
volumeBindingMode: Immediate
Create the gp2-immediate storage class:
kubectl apply -f gp2-immediate.yaml
```

Running the tests

In this section, we list the steps to run the VMWare Weathervane benchmark on the clusters under test. We used the findMaximumSingleRun test type in order to determine the largest number of WvUsers that each cluster could sustain while still maintaining the QoS standards required to pass the benchmark.

Run the tests

1. Log into the run-harness instance.
2. Obtain the `kubeconfig` file:

```
sudo aws eks --region us-east-1 update-kubeconfig --name eks-8cpu
```

3. Navigate to the weathervane folder.
4. Open the run configuration file with the following command:

```
sudo vim weathervane.config.k8s.quickstart
```

5. Ensure the following information is correct:
   a. description: enter a description of the run
   b. configurationSize: small2
   c. runStrategy: findMaxSingleRun
   d. kubeconfigFile: [kube config file location]
   e. kubeconfigContext: [k8s cluster name]
   f. appIngressMethod: clusterip
   g. StorageClasses: gp2-immediate
   h. numAppInstance: [# of app instances]
   i. useAppServerLimits: false
6. Exit the text editor.
7. Run the test:

```
sudo ./runWeathervane.pl –configFile=weathervane.config.k8s.quickstart
```

8. Once the test completes, delete the PVCs created during the run:

```
sudo kubectl delete pvc --selector=app=auction --all-namespaces
```

9. Repeat the test two more times for a total of three runs. Report the results of the median run.

Read the report at http://facts.pt/sAOVfuj ▶

This project was commissioned by Intel.