Deploying MQ to the Cloud

Matthew Whitehead
IBM MQ Development
mwhitehead@uk.ibm.com

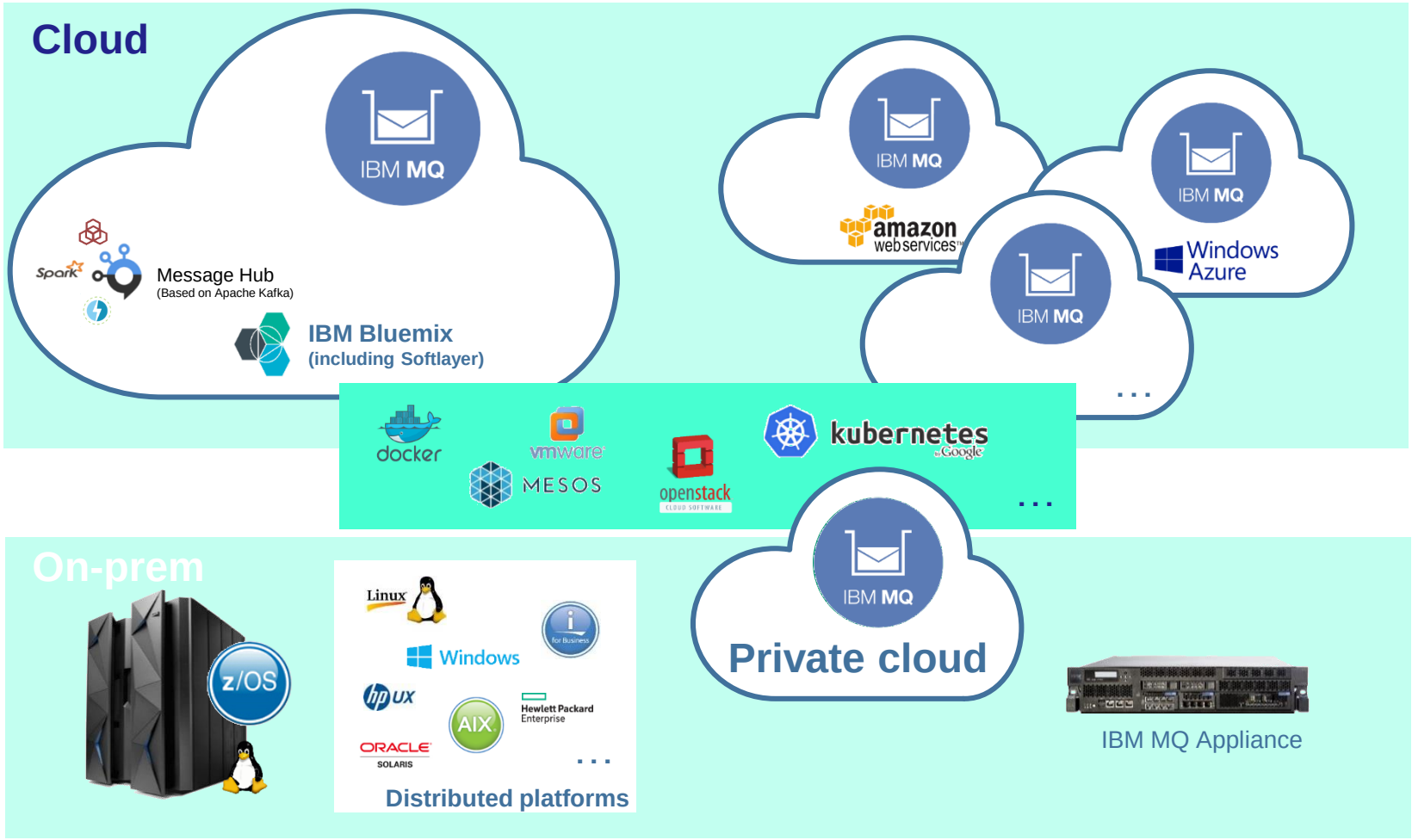
Agenda

- MQ in the Cloud
- Types of Cloud Environment
- MQ & other IBM Messaging Solutions
- Cloud Deployment Considerations

Why Cloud?

- Doing more with less
- Being more ready to change
- Making the development process less heavyweight
- Paying for what you use
- Integrating with other cloud services
- Rapidly scaling up and down with demand

MQ is everywhere



MQ, ready for the cloud

MQ's capabilities were ready for the cloud before the cloud was even a *thing*

- Dynamic client connectivity
- Dynamic scaling
- Workload balancing
- High security
- High scale and robustness
- Repeatable and remote administration and monitoring
- Perfect for integrating systems across any cloud

…

But are you using it that way?

Rethink MQ

Don’t confuse old MQ practices for MQ itself

“MQ is too hard to use”

“Our MQ system is too complicated to change”

“MQ isn’t cloud, it’s too old!”

How many of these do you have?

- Hand crafted, shared queue managers
- Applications hard coding connection details
- Applications bound to a single IP address
- Edge security at most
- Internal architecture complexity exposed to the applications
- A lengthy change control process
- Manual installation, deployment and configuration

Rethink how you use MQ!

MQ can do anything – but you don't have to use it all!

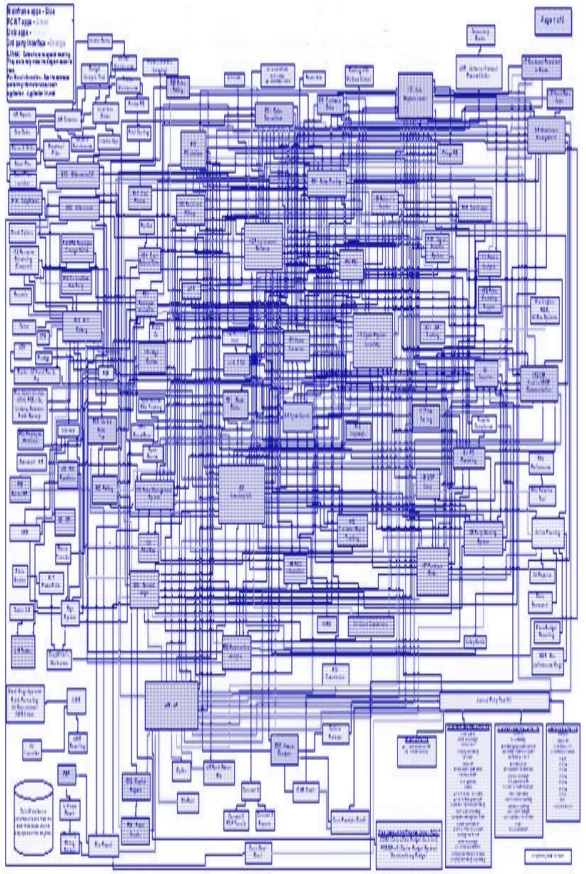
MQ has near infinite variations in architectures and configurations

Excellent for allowing customisation to your company's exact requirements
Bad if you try to use a bit of everything

To support cloud deployments, limit those variations and customisations to a minimal set of *patterns*

Having a set of patterns enables automatable and repeatable deployments to easily push into the cloud
It also simplifies the job of the operations teams by minimising per-system complexity

You may need to compromise on some of your customisations and optimisations



Which type of cloud?

IaaS? PaaS? CaaS? FaaS? SaaS?

Not made up

Which Type of Cloud?

- **IaaS** (Infrastructure-as-a-Service - VMs)
  - Are good for large services/apps, but generally not ideal
  - May be used more like physical machines, but with added flexibility
- **CaaS** (Containers-as-a-Service - e.g. Kubernetes)
  - Are good for micro-services/apps
  - Potentially quite short-lived
- **PaaS** (Platform-as-a-Service - e.g. Bluemix, Cloud Foundry)
  - Are great for application code in general
  - Handing off infrastructure worries to someone else
- **FaaS** (Functions-as-a-Service - e.g. OpenWhisk, AWS Lambda)
  - Could be used for occasional compute loads
  - Will likely drive lots of short-lived connections, so may not perform well for some messaging workloads
  - Most support JavaScript (could use the MQ Light API), but some can support Java, C# and more

Which Type of Cloud?

QMs

- **IaaS** (Infrastructure-as-a-Service - VMs)
  - Are good for large services/apps, but generally not ideal
  - May be used more like physical machines, but with added flexibility

QMs

Apps

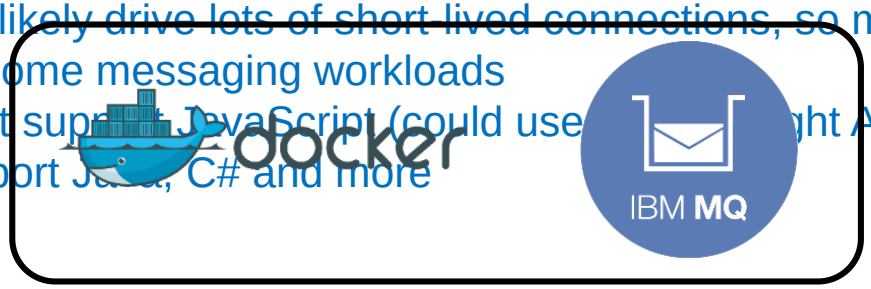
- **CaaS** (Conta… …ce - e.g. Kubernetes)
  - Are goo… …apps
  - Potenti…
  - e.g.

Apps

- **PaaS** (Platform-as-a-Service - e.g. Bluemix, Cloud Foundry)
  - Are great for application code in general
  - Handing off infrastructure worries to someone else

Apps

- **FaaS** (Functions-as-a-Service - e.g. OpenWhisk, AWS Lambda)
  - Could be used for occasional compute loads
  - Will likely drive lots of short-lived connections, so may not perform well for some messaging workloads
  - Most sup… JavaScript (could use … API), but some can support J…, C# and more
  - e.g.



Which Cloud Provider(s)?

- EC2
- EBS/S3/EFS
- Lambdas
- Cloudwatch
- …

- VMs
- App Service
- Active Directory
- IoT Hub
- Visual Studio Services
- …

- VMs
- OpenWhisk
- BluemixContainer Service
- CloudFoundry
- Logmet
- …

- Compute Engine
- App Engine
- Container Engine
- Cloud Functions
- BigQuery
- …

Poll Time!

- Show of hands for:
  - Virtual Machines (EC2, Bluemix VMs, Azure VMs, On-prem etc.)
  - Containers (Docker, AWS ECS, Bluemix Containers etc.)
  - Hybrid Messaging (Linking on-prem messaging to cloud-apps)
  - Serverless Compute (OpenWhisk, Lambdas etc.)

Deploying & Orchestration

MQ on OpenStack, part one: Creating an image using Packer

Arthur Barr | *Apr 6 2016* | *Visits (5798)* 5

MQ on OpenStack, part two: Managing an MQ environment using Heat

Arthur Barr | *Apr 20 2016* | *Visits (3453)* 4

MQ on OpenStack, part three: Automated client connection PoC using MQ v9 CCDT URL feature.

RobParker | *Aug 17 2016* | *Comment (1)* | *Visits (2706)* 1

MQ in Docker is now supported for production use

Arthur Barr | *Nov 30 2015* | *Comments (2)* | *Visits (8606)* 3

Basic deployment of MQ on AWS

Arthur Barr | *May 25 2016* | *Visits (7515)* 5

https://developer.ibm.com/messaging/mq-on-cloud/

Deploying & Orchestration

Provisioning an Environment

- Create VM, start docker container etc.
- Setup virtual network interface
- Create storage
- Create security policies

Installing MQ

- Initial installation
- Applying fixpacks
- Migrating between versions
- Managing licenses

Running queue managers

- Create/start/stop/delete QM
- Naming queue managers
- Binding apps to QMs
- Failing over

Deploying & Orchestration

Provisioning an Environment

- Create VM, start docker container etc.
- Setup virtual network interface
- Create storage
- Create security policies

Installing MQ

- Initial installation
- Applying fixpacks
- Migrating between versions
- Managing licenses

Running queue managers

- Create/start/stop/delete QM
- Naming queue managers
- Binding apps to QMs
- Failing over

Deploying & Orchestration

Provisioning an Environment

- Create VM, start docker container etc.
- Setup virtual network interface
- Create storage
- Create security policies

Installing MQ

- Initial installation
- Applying fixpacks
- Migrating between versions
- Managing licenses

Running queue managers

- Create/start/stop/delete QM
- Naming queue managers
- Binding apps to QMs
- Failing over

Deploying & Orchestration

Provisioning an Environment

- Create VM, start docker container etc.
- Setup virtual network interface
- Create storage
- Create security policies

Installing MQ

- Initial installation
- Applying fixpacks
- Migrating between versions
- Managing licenses

Running queue managers

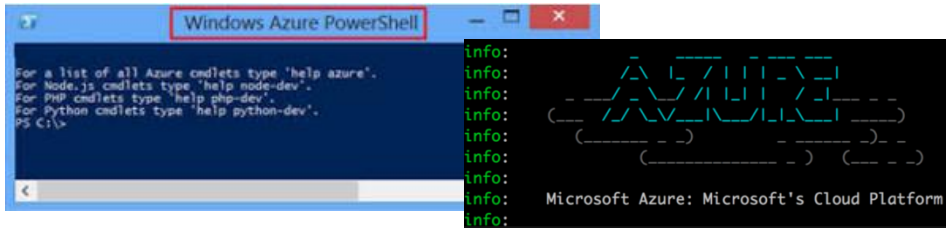
- Create/start/stop/delete QM
- Naming queue managers
- Binding apps to QMs
- Failing over

+

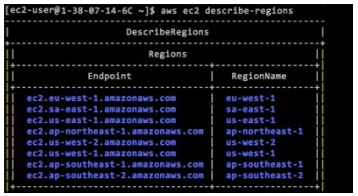
Manual configuration & management tools

IBM Bluemix™

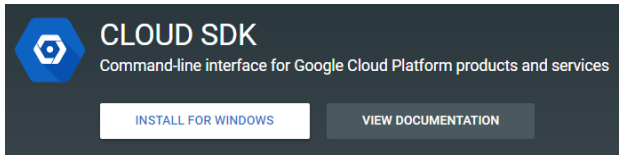
Microsoft Azure



amazon web services



Google Cloud Platform



Resource & environment management

- Cross cloud deployment & configuration
- Different provider options (Azure, AWS, OpenStack, Google Cloud, SoftLayer)

- AWS orchestration framework
- Configure networks, VMs, storage etc.

- Define resources, security, storage etc. in a template
- Used to orchestrate OpenStack deployments

https://developer.ibm.com/messaging/mq-on-cloud/

Installing MQ

- Create immutable OS images with MQ pre-installed
- Cookie-cutter installations of e.g. MQ & Redhat
- Deploy images to chosen IaaS service

- Provides an MQ cookbook to include in Chef recipes
- Use to perform repeatable & automatable MQ installations

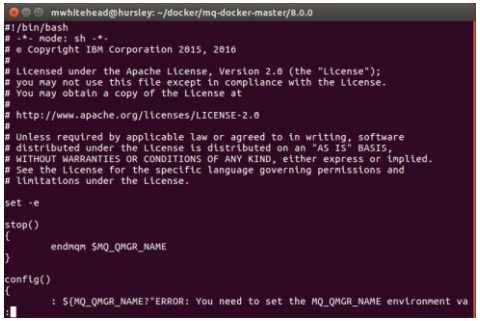
- Manage and script software installation
- Similar to Chef, use in conjuction with an MQ install package to automate and repeat installations

https://developer.ibm.com/messaging/mq-on-cloud/

Creating & starting QMs

- As well as being able to install MQ, can create and start queue managers
- Can combine with installation steps to give a complete MQ environment

- Simple shell script can be a useful lowest common denominator
- Use in a heat template, docker image, VM…
- Not dependent on specific framework

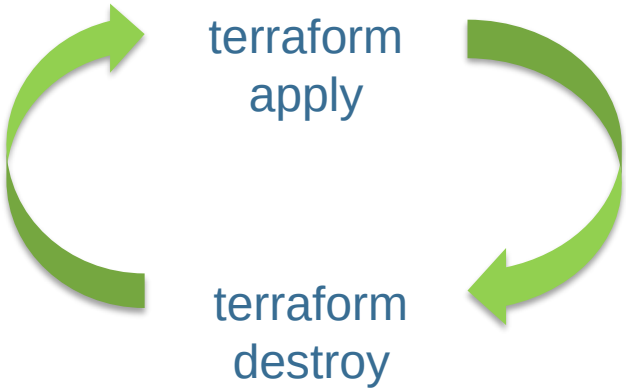


https://developer.ibm.com/messaging/mq-on-cloud/

Other benefits of orchestration – tear-down

Deleting resources when they're no longer needed

- Orchestration isn't limited to creating resources
  - Tear down resources when they're no longer required
  - Quickly spin up everything necessary to run a scenario, tear everything down when it's no longer needed
  - Useful for running one-off tests, creating short-lived development environments
  - Keeps cloud costs to a minimum – only pay for what you need at the time
  - Don't need to manually keep track of what you created
- Repeatability



https://developer.ibm.com/messaging/mq-on-cloud/

Containers

Resource & environment management

Installing MQ

Creating & starting QMs

(See **MQ in Containers** sessions: Wednesday 3.50pm)

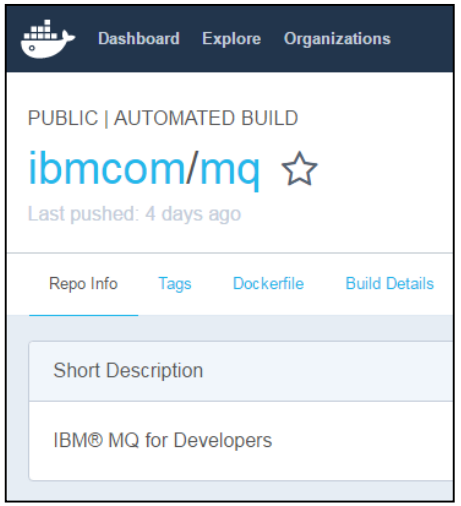
Containers

- Containers provide a similar environment to a VM but lighter in weight
  - A virtual machine provides an abstraction of the physical hardware
  - A container abstracts the OS level, typically at the user level
- Linux containers
  - Containers all share the same OS kernel
  - Images are constructed from layered filesystems
  - Containers isolate applications from each other and the underlying infrastructure

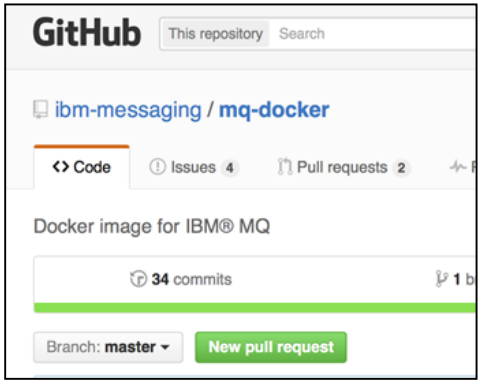
MQ Docker Container

- MQ 8.0.0.4+ supported to run inside a Docker image
  - Details: https://ibm.biz/mqdocker
- Brings the benefits of Docker to MQ
  - Lightweight containers for running MQ
  - Predictable and standardized units for deploying MQ
  - Process, resource and dependency isolation
- IBM samples for customizing and building your own Docker images
  - Runs an MQ queue manager inside a container, isolated from the rest of your system

Binary image in Docker Hub



Source in GitHub




```
mwhitehead@mrw-ubuntu-1604: ~
mwhitehead@mrw-ubuntu-1604:~$ docker pull ibmcom/mq
Using default tag: latest
latest: Pulling from ibmcom/mq

71a21fdea81d: Downloading 13.52 MB/65.7 MB
cf68a3ea6e1d: Download complete
31cb2a4d344a: Download complete
0341b6fcb0fe: Download complete
```

MQ Docker Container

- Consideration needs to be given to:
  - Where /var/mqm data goes when the container stops
  - How to name queue managers
  - Changing channel definitions with updated IP address
    - In many container environments a re-provisioned container is given a new IP address
  - How you approach scaling down
  - The difference between more long lived containers (perhaps running full repositories) and short lived containers
- May be useful simply for basic, on-prem scenarios to reduce complexity

IBM Messaging Solutions

MQ	MessageHub	MessageSight and Bluemix IoT
Enterprise features	Message streams	IoT scenarios
24x7, 365	Based on Kafka	MQTT protocol
Multi-platform (z/OS, IBM I, Unixes, Windowses)	Very high throughput, highly scalable	Good for very high client numbers (10,000s)
Rich feature set (lots of dials and options)	Ideal for Bluemix micro-services	
Some static components (e.g. full repositories)	Currently Bluemix only	

IBM Messaging Solutions – Cloud Use-Cases

- MQ
  - Hybrid messaging – cloud to/from on-prem
  - Inter-cloud communication
  - Deployable anywhere that can host VMs or containers

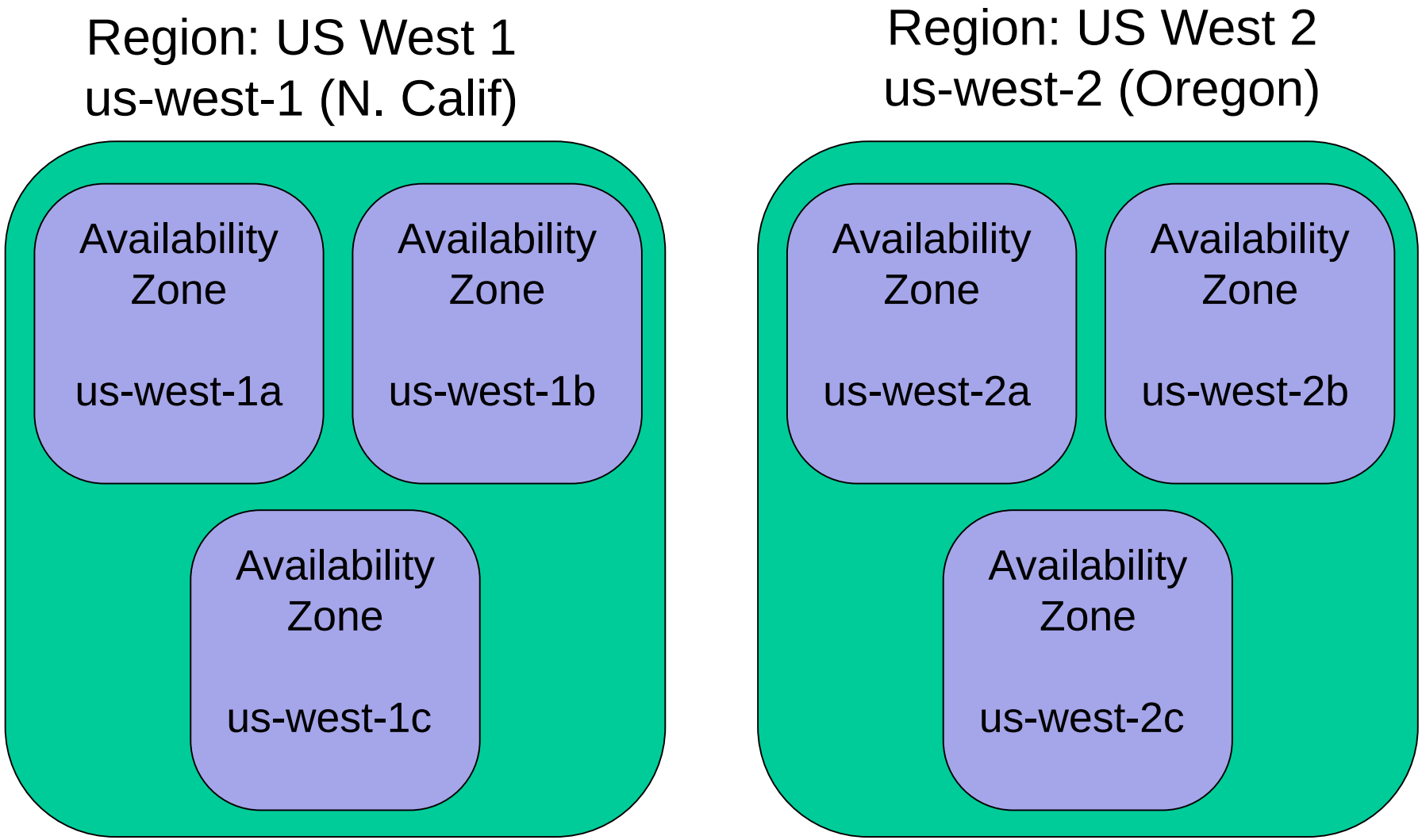
- MessageHub
  - Ideal for micro-service architectures
  - Streaming/Real-time analytics
  - Bluemix/Softlayer only

- MessageSight and IoT
  - Ideal for connecting huge numbers of devices
  - Low-bandwidth network scenarios (e.g. mobile)

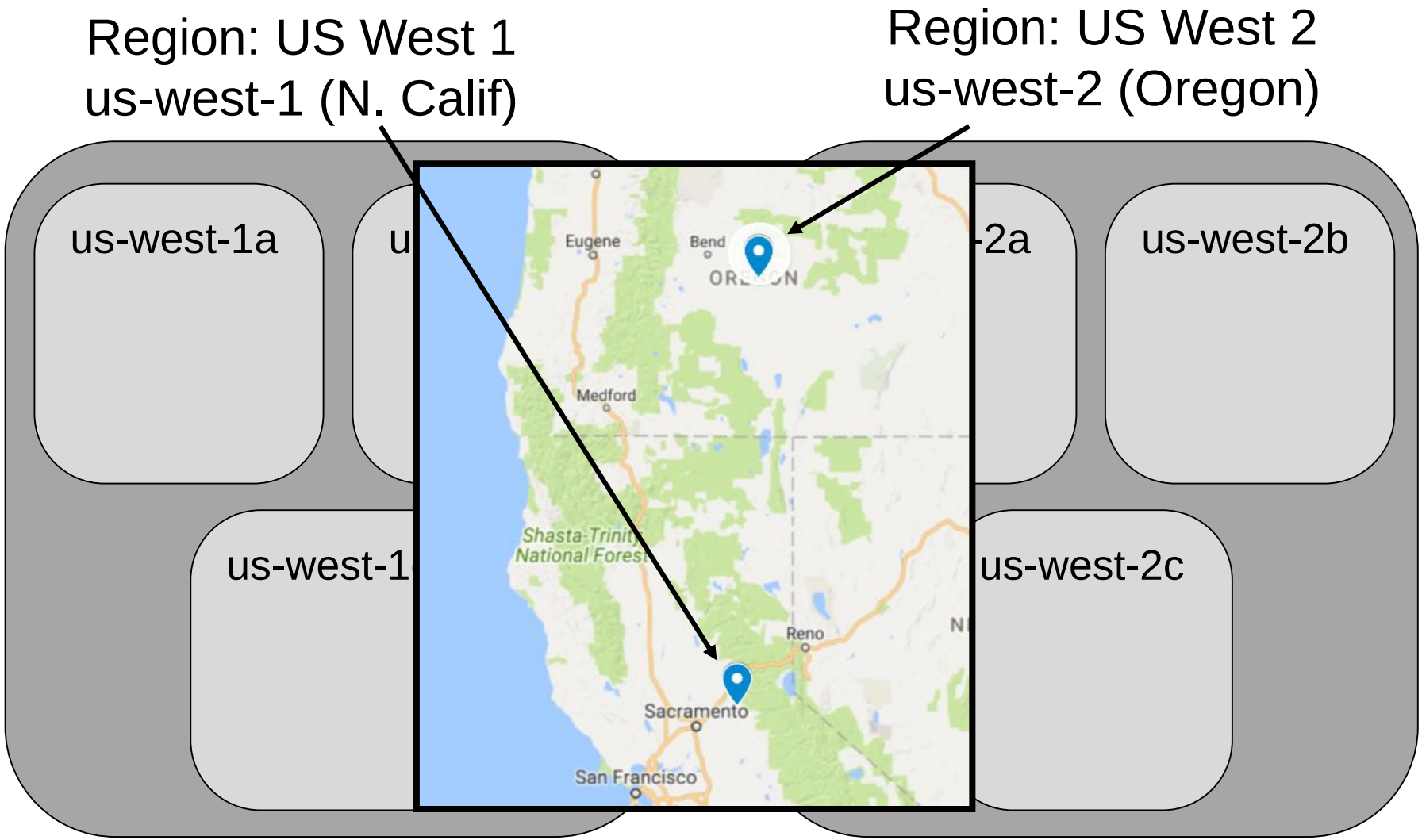
Persistent Storage Considerations

- **Reliability of storage**
  - Replicated across failure domains / availability zones?
  - Are disk writes cached?
  - What's the failure rate of disks?
- **Connecting to the right persistent storage**
  - When a queue manager's compute resource is moved (e.g. run a container in a different VM), then something needs to connect the queue manager to the correct storage.
  - e.g. the correct block storage volume, or directory on networked file storage.
- **Identifying the right persistent storage**
  - A very basic cloud orchestration setup could result in multiple instances of "QM1"

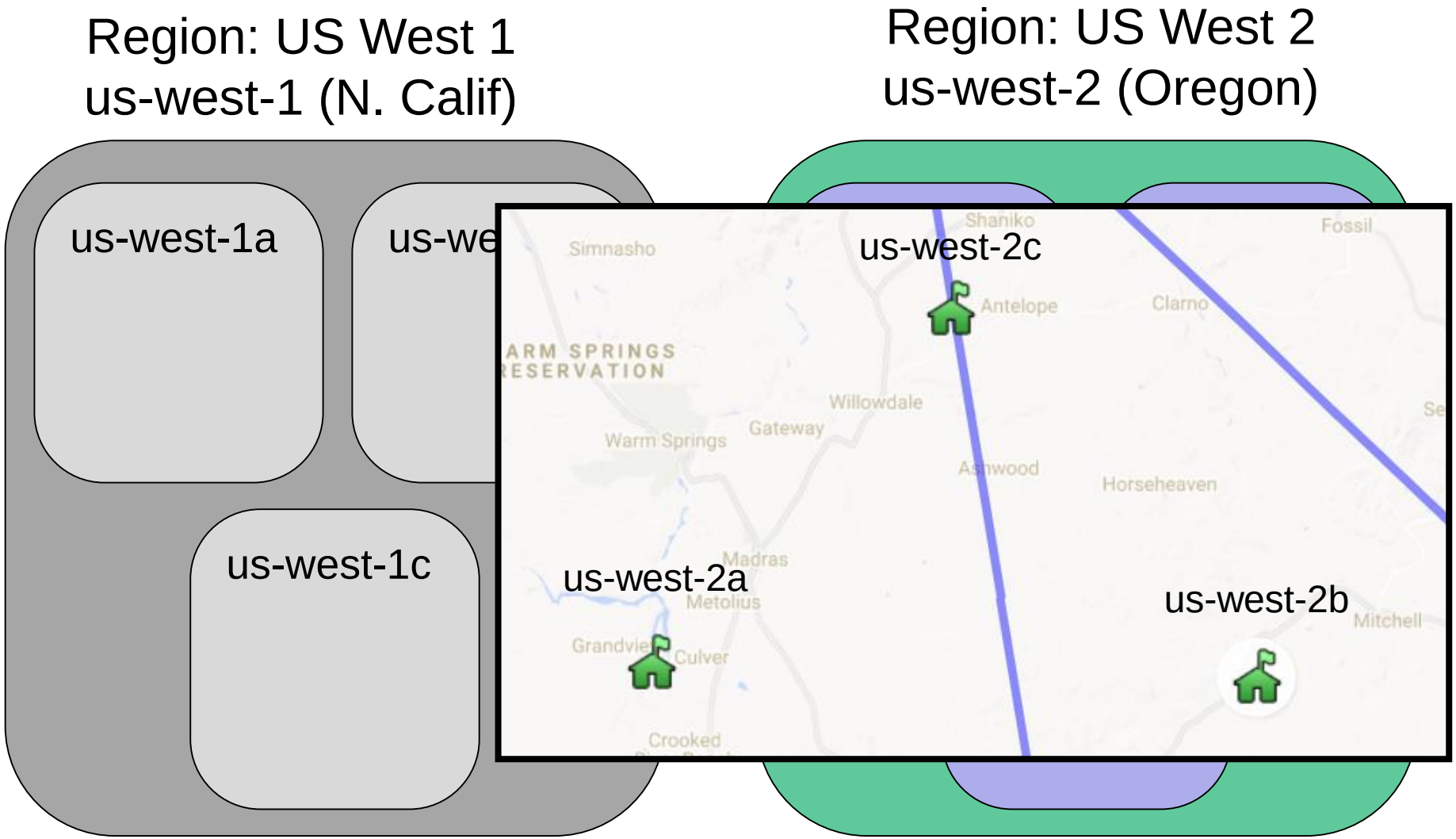
Persistent Storage Considerations



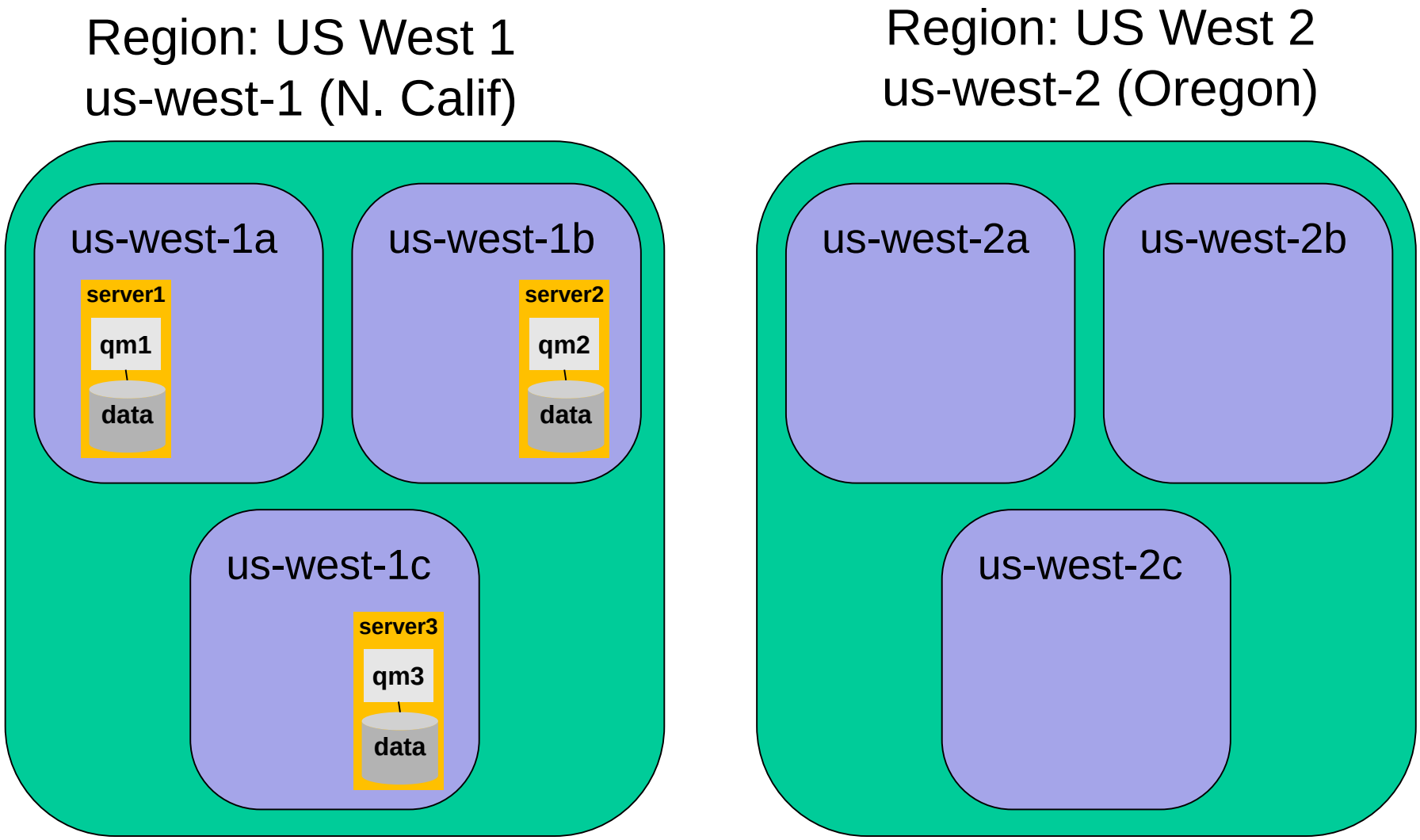
Persistent Storage Considerations



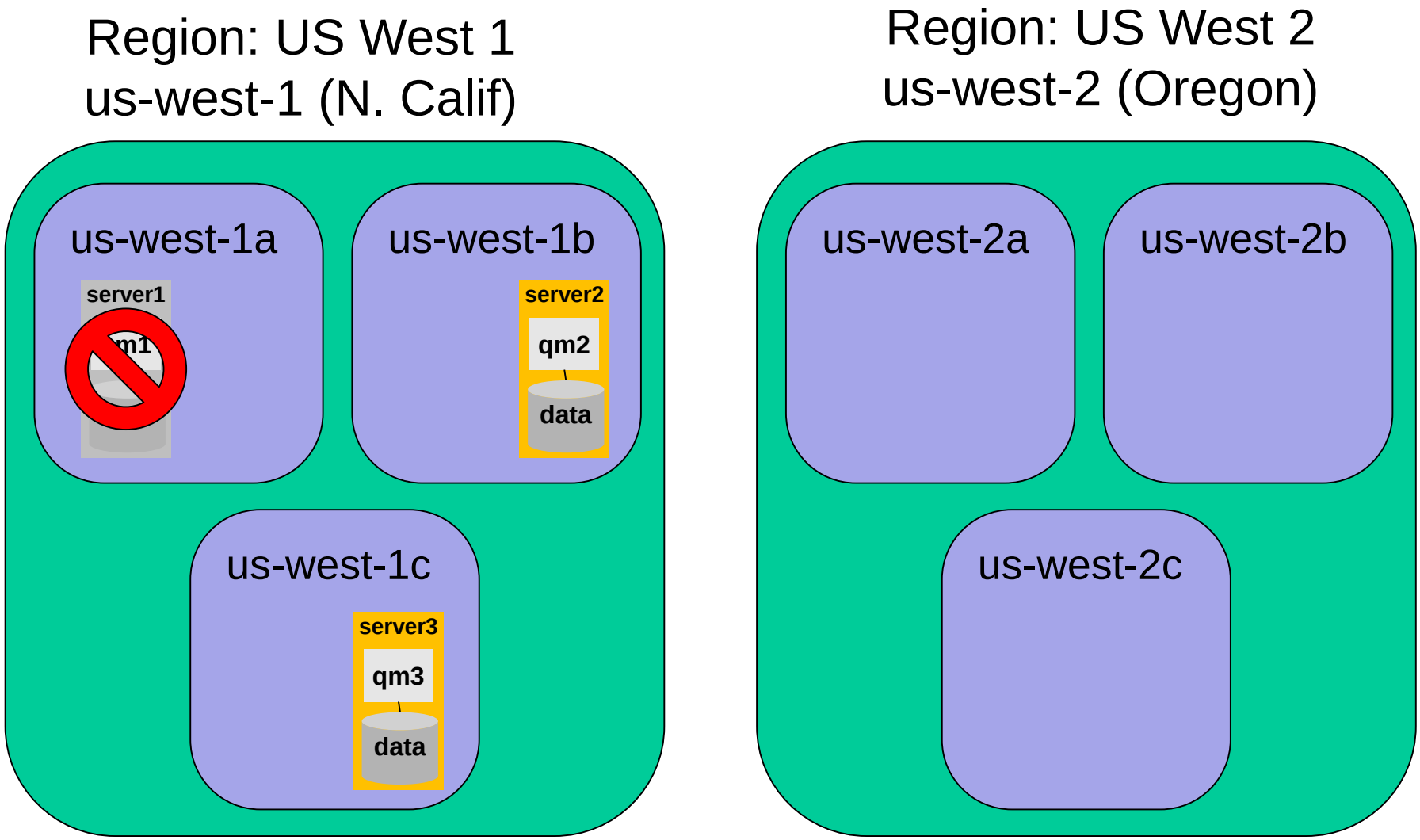
Persistent Storage Considerations



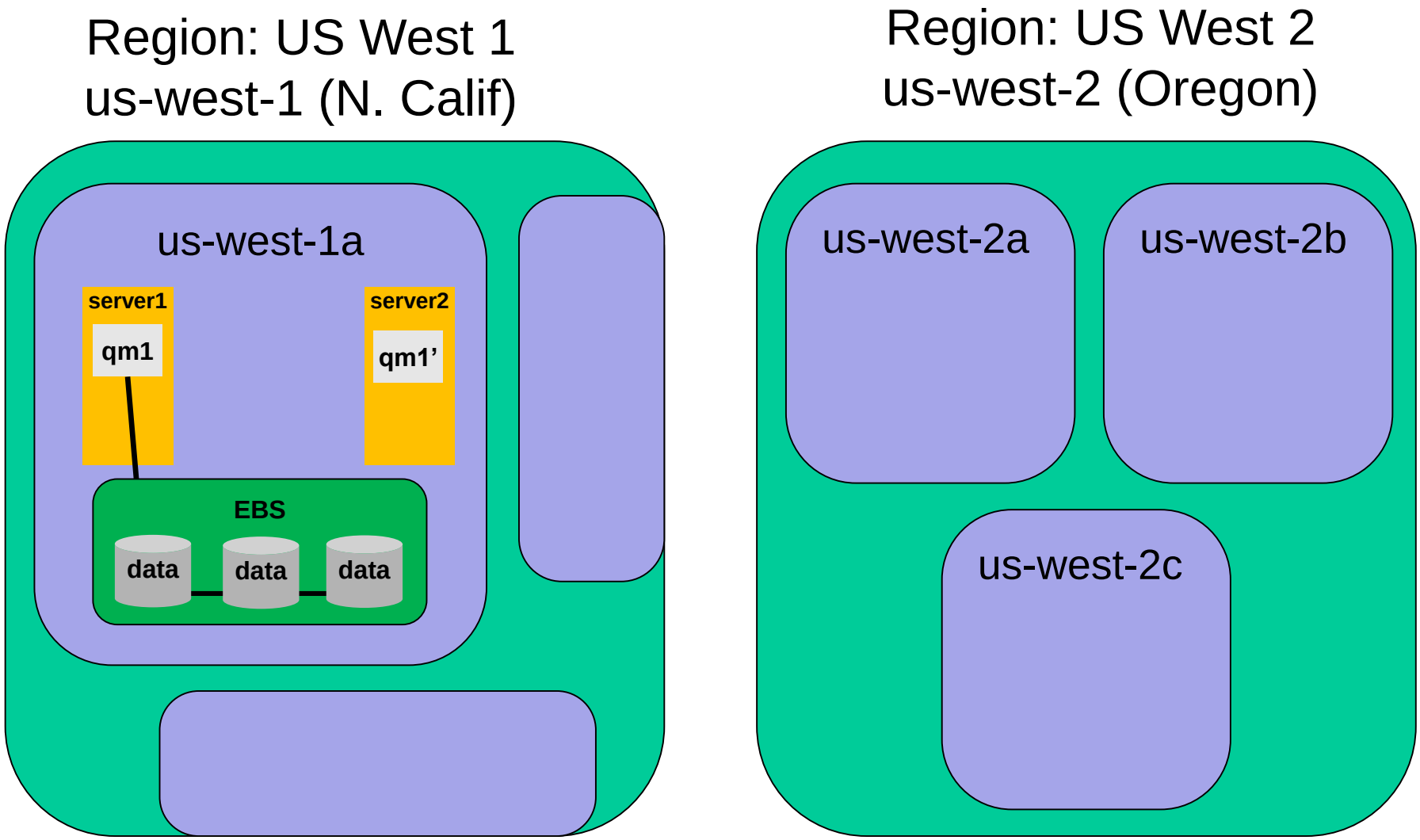
Persistent Storage Considerations – Local SSDs



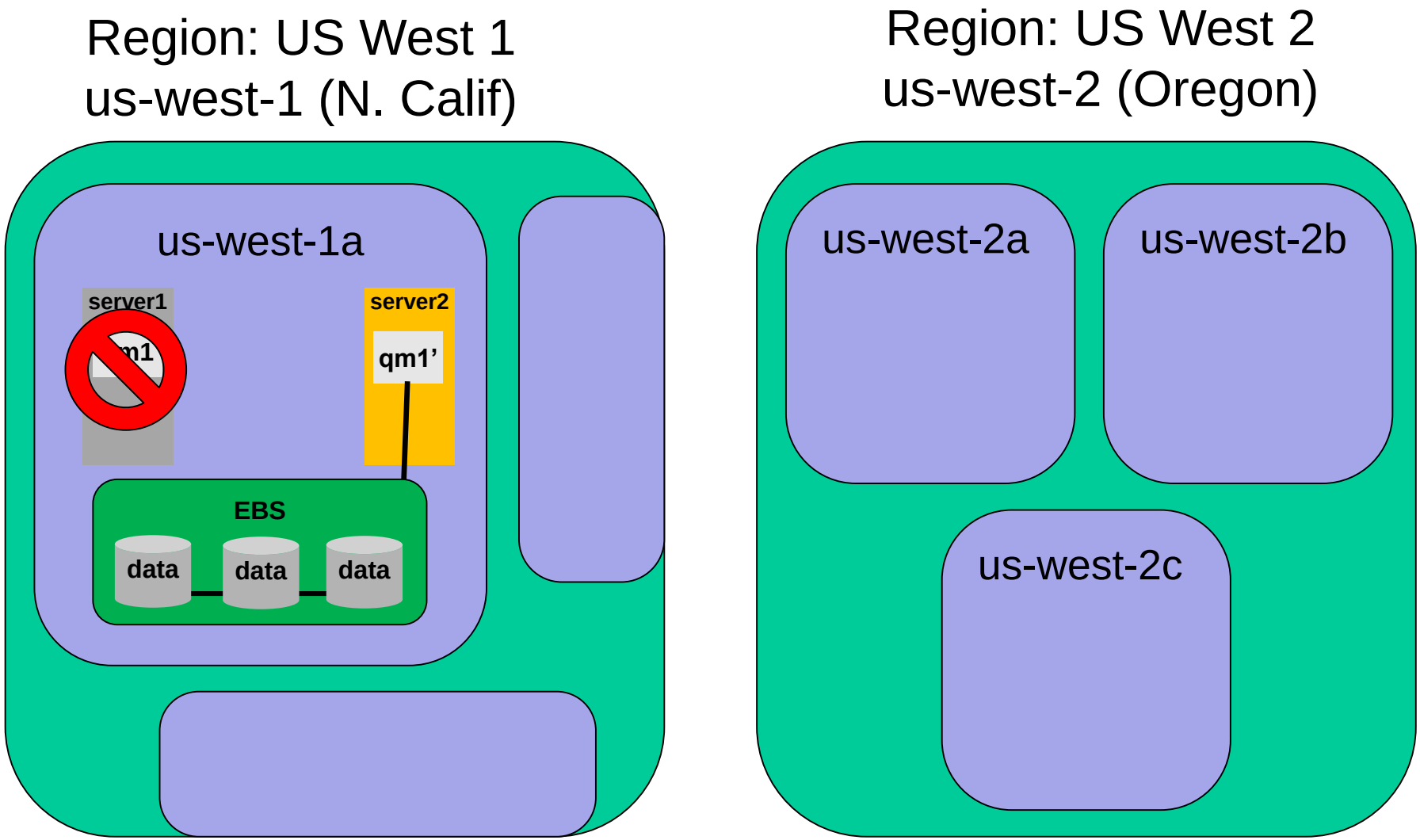
Persistent Storage Considerations – Local SSDs



Persistent Storage Considerations – Elastic Block Storage

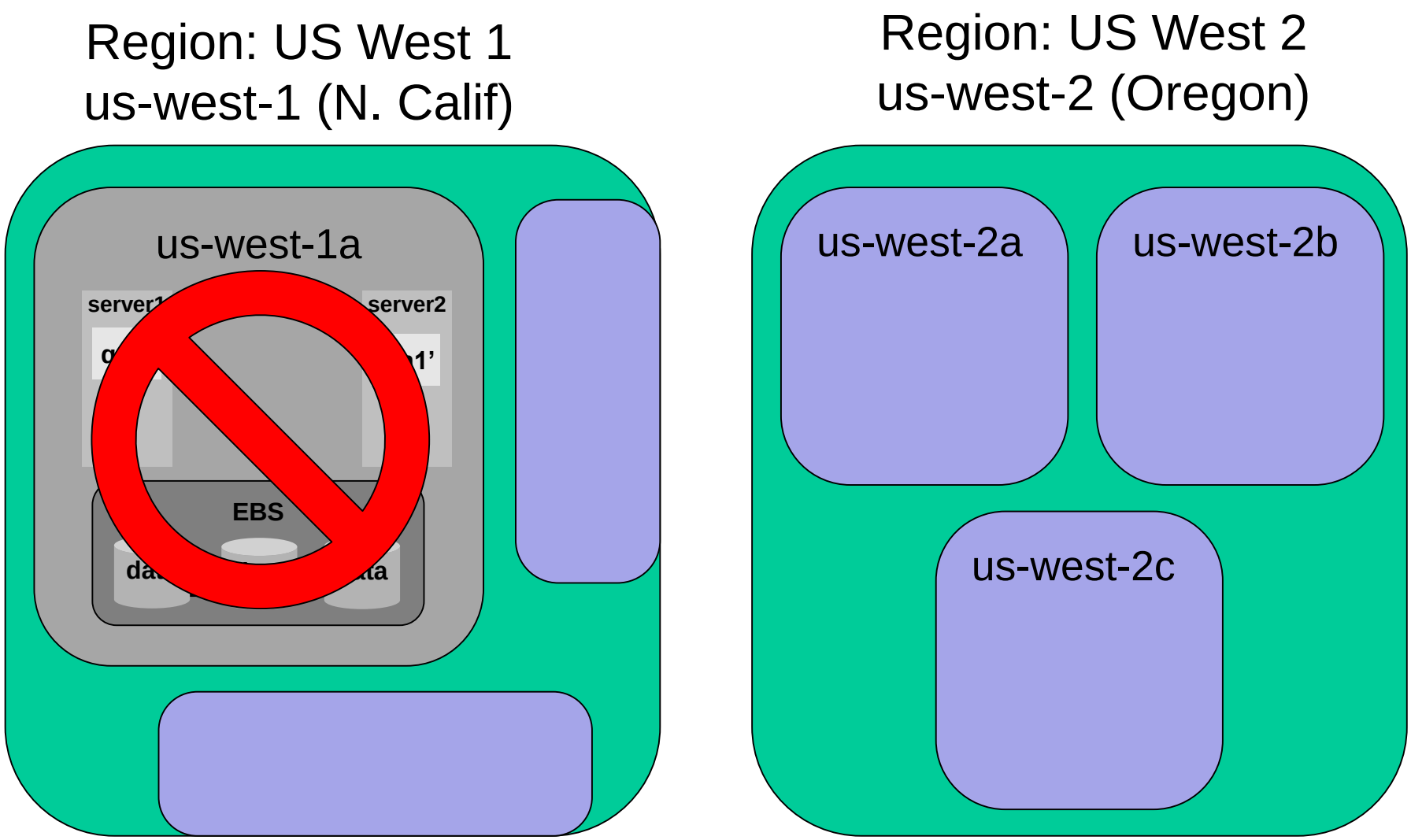


Persistent Storage Considerations – Elastic Block Storage



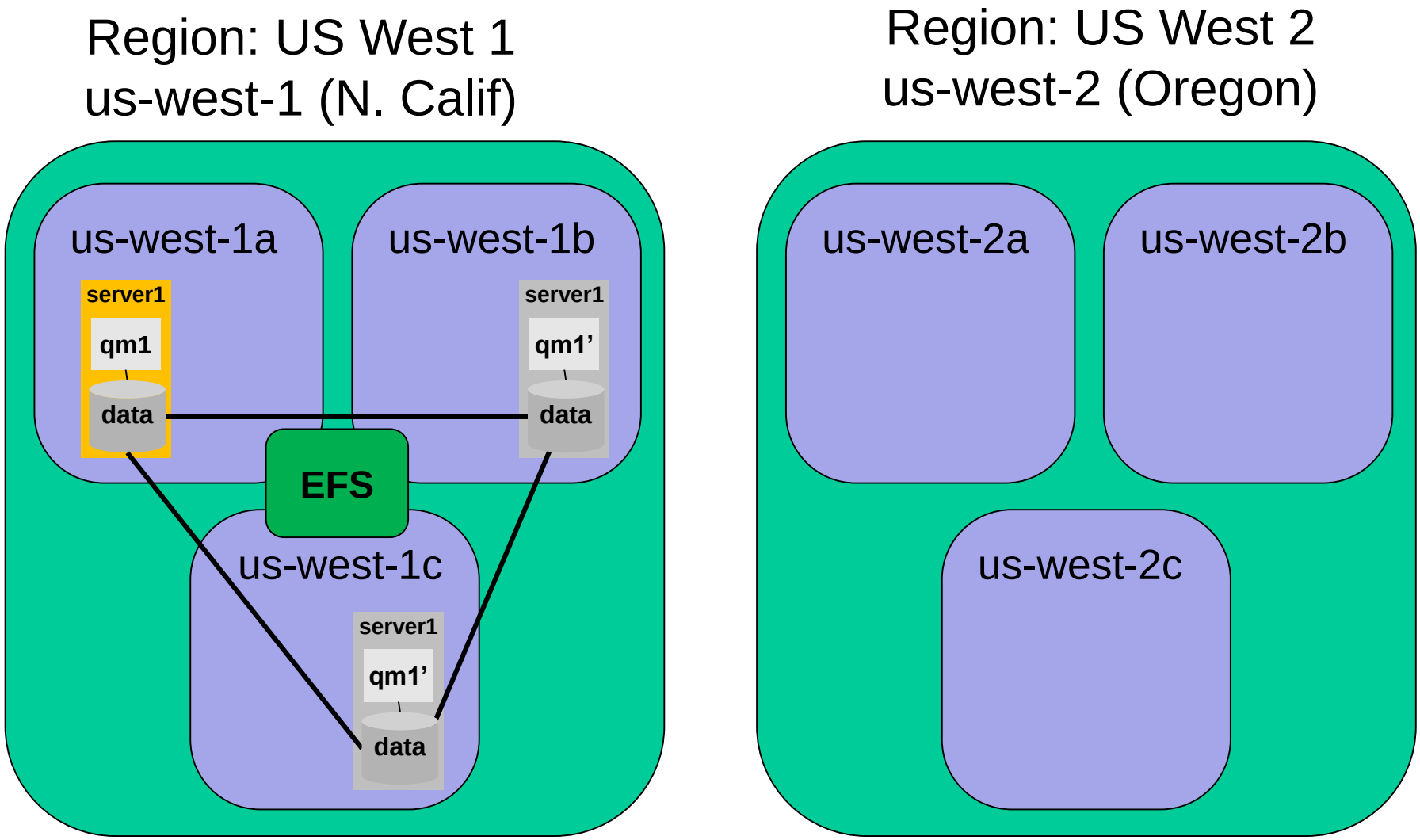
- MQ multi-instance/cloud provider auto-restart/custom control e.g. Pacemaker

Persistent Storage Considerations – Elastic Block Storage

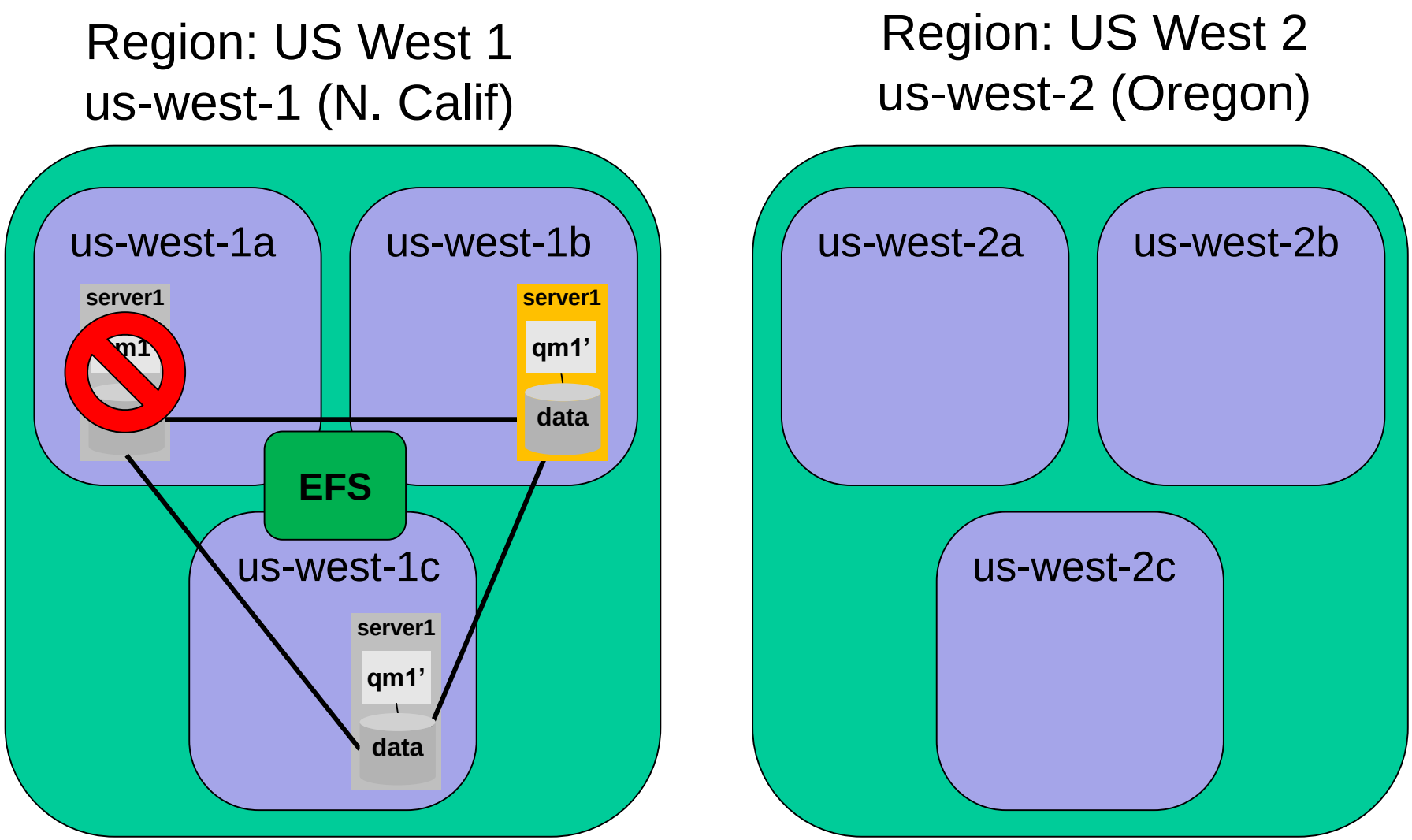


- MQ multi-instance/cloud provider auto-restart/custom control e.g. Pacemaker

Persistent Storage Considerations – Elastic File Storage



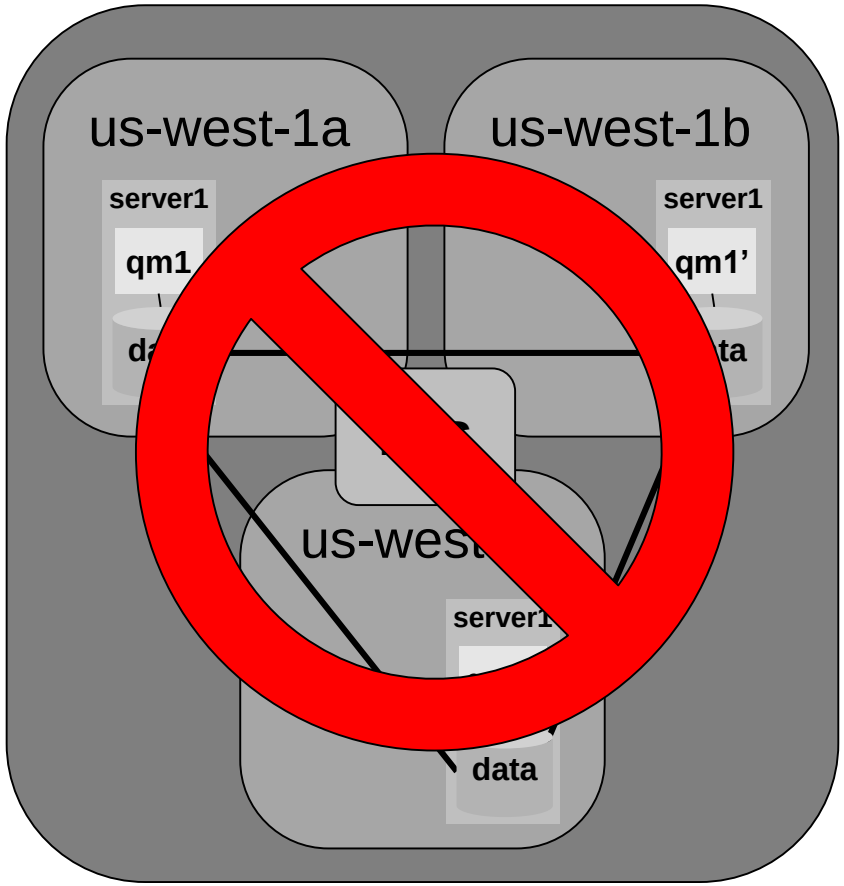
Persistent Storage Considerations – Elastic File Storage



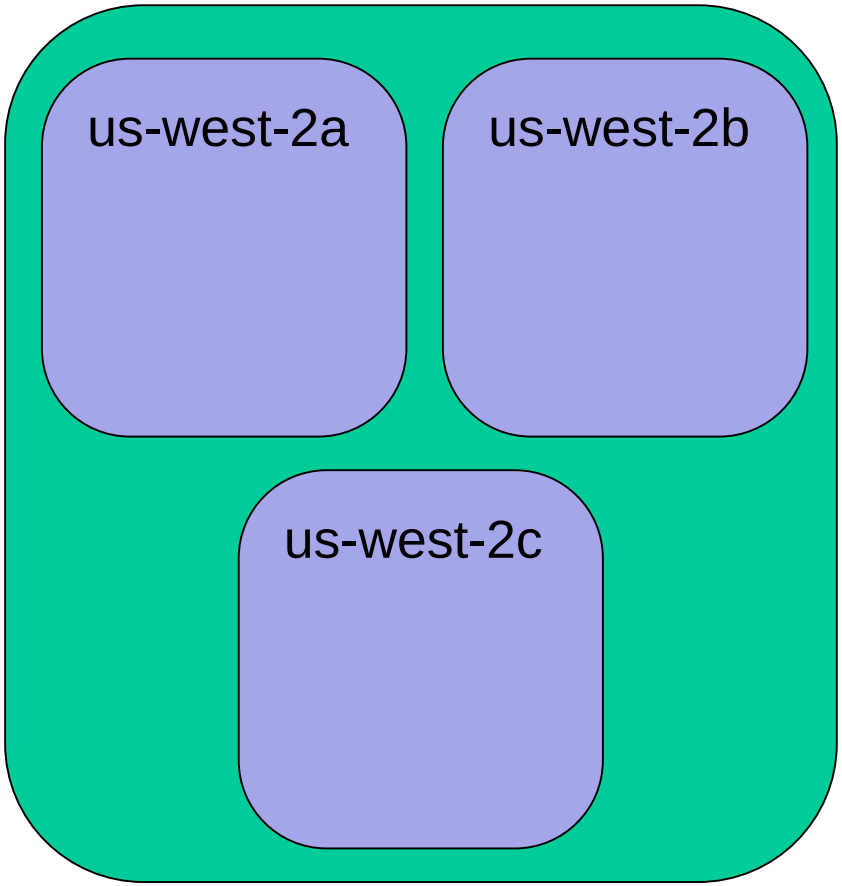
- Cloud provider auto-restart/custom control e.g. Pacemaker (**not** MQ multi-instance)

Persistent Storage Considerations – Elastic File Storage

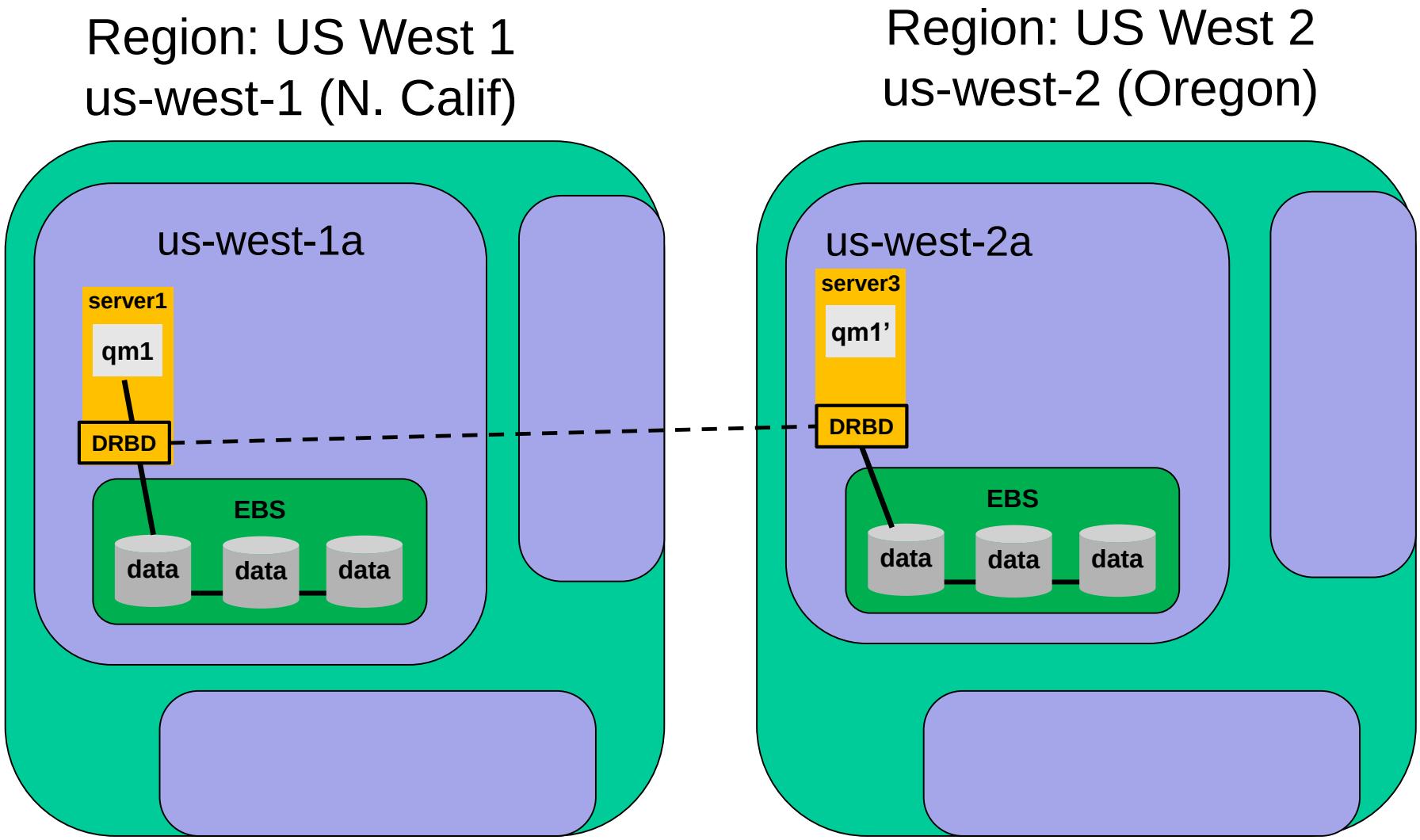
Region: US West 1
us-west-1 (N. Calif)



Region: US West 2
us-west-2 (Oregon)

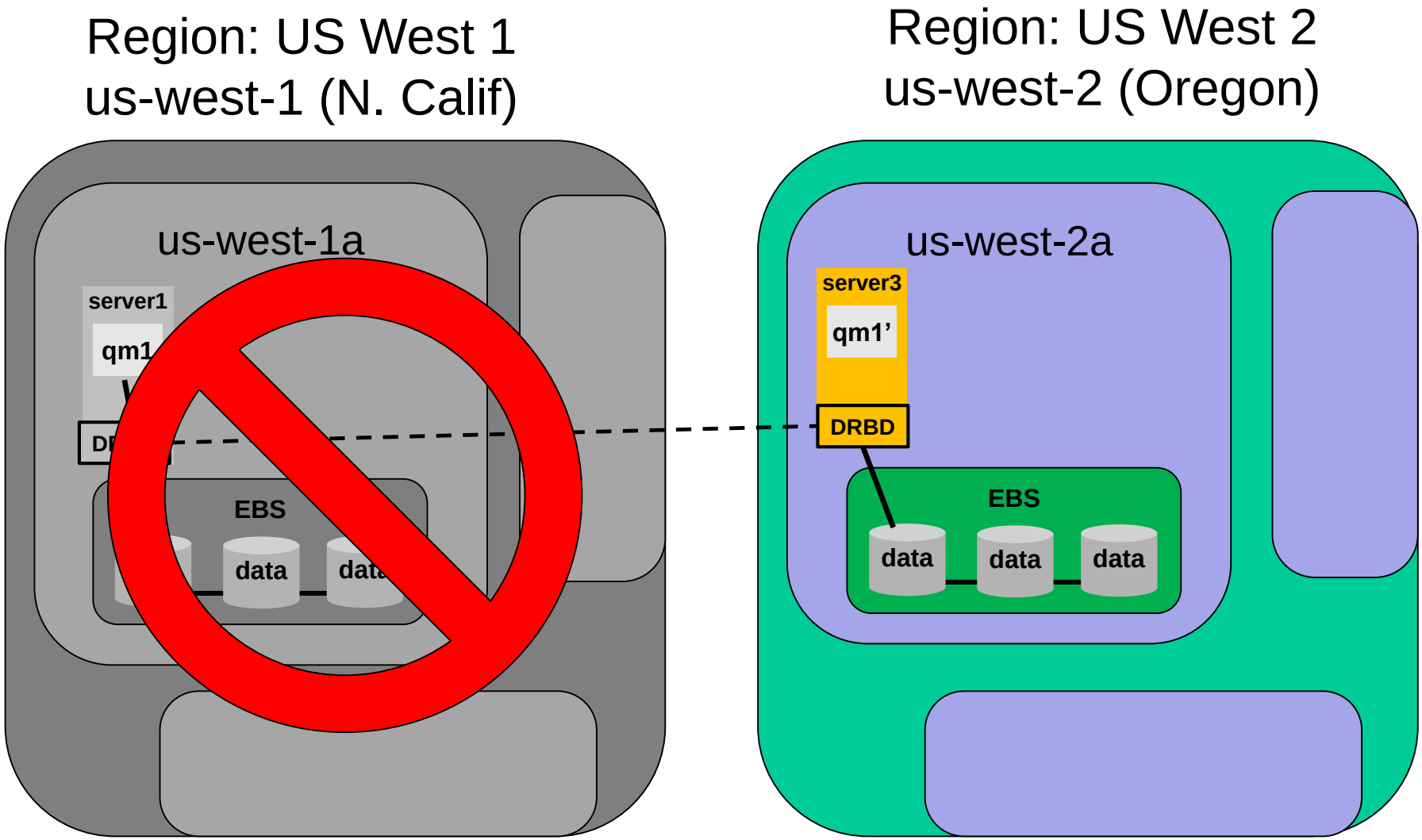


Persistent Storage Considerations – Elastic Block Storage



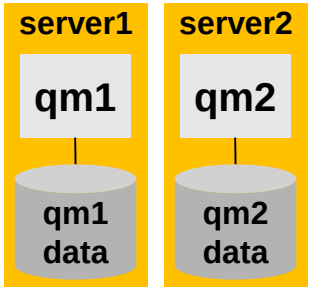
- DR rather than HA - asynchronous replication so some messages at risk of loss

Persistent Storage Considerations – Elastic Block Storage



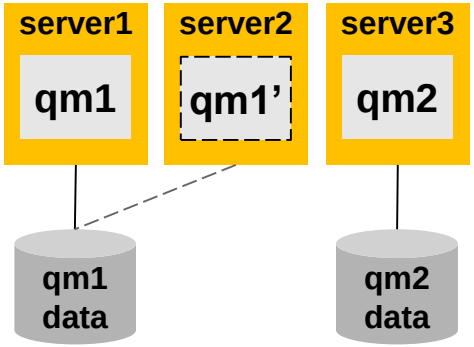
Persistent Storage - Local

- Local storage typically has the same life span as compute resource (e.g. VM or bare metal server)
- Often very fast to access local storage
  - SSDs
- Containers typically have a short life span, usually making local storage an unsuitable option for MQ
- This may be an option for long-lived bare metal servers



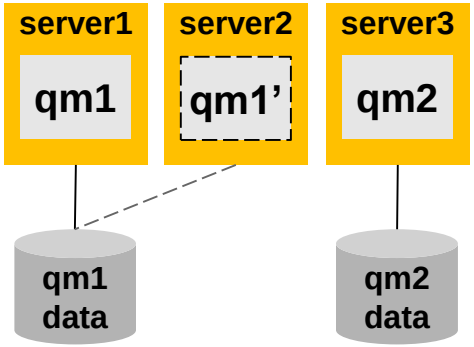
Persistent Storage – Networked Block Storage

- For example: OpenStack Cinder, Amazon Elastic Block Storage, Ceph, DRBD
- You can use well-tested filesystems
- Performance needs to be considered
- Key challenge: something needs to re-attach the block storage to a different VM if the queue manager is moved (e.g. because of failure, or VM image update)



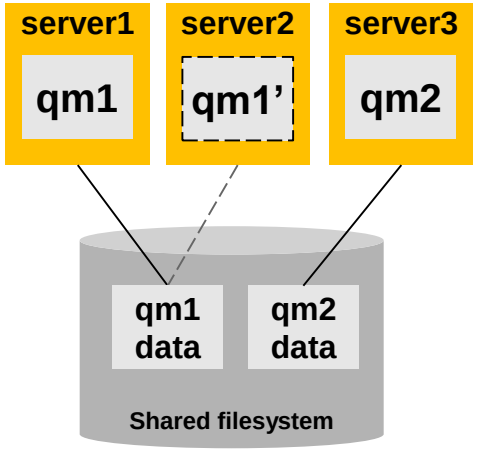
Persistent Storage – Networked Block Storage

- Can be used to implement MQ HA and/or DR
- Block storage under /var/mqm synchronised to an alternative availability zone
- Something must monitor the state of the primary queue manager…
  - E.g. Pacemaker
- …and then mount the block storage at the standby availability zone before starting the backup queue manager
- DR can be achieved by manually synchronising to an alternative region
  - Using e.g. DRDB

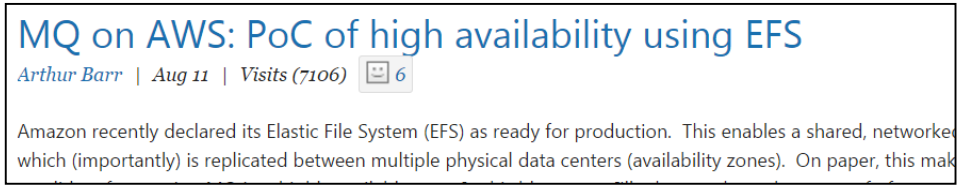


Persistent Storage – Networked File System

- **Examples: GPFS, NFS, Amazon EFS**
- **Key challenge: MQ is sensitive to filesystem characteristics such as locking**
  - For example, NFS V3 is known not to work
  - IBM publishes a list of filesystems tested with MQ: http://www-01.ibm.com/support/docview.wss?uid=swg21433474
- **Not just for multi-instance queue managers – can easily handle the case where a queue manager is moved.**
- **Performance needs to be considered**



MQdev article discusses MQ on EFS


MQ on AWS: PoC of high availability using EFS

Arthur Barr | *Aug 11* | *Visits (7106)* 6

Amazon recently declared its Elastic File System (EFS) as ready for production. This enables a shared, networke
which (importantly) is replicated between multiple physical data centers (availability zones). On paper, this mak

QM Availability Decisions

(As an example, AWS commit to 99.95% uptime in an availability zone)

- Single instance with failure detection and automatic restart
    - Data may or may not be safe (depending on cloud provider) but only available to new instance in the same availability zone
- Active-Passive, warm instance waiting in another availability zone to take over in event of failure
    - MQ multi-instance (using e.g. synchronized block storage), or…
    - Cloud auto-failover (e.g. AWS EC2 with AWS EFS)
- Active-Active, workload balancing between instances in multiple availability zones
    - MQ clustering fits well into this architecture, or…
    - MQ client CCDT workload balancing, or…
    - Single instances, use cloud-technologies to workload balance

Availability – Pros and Cons

Single instance, automatic restart

- Higher performance – no or local-only data synchronisation ✓
- Simple architecture ✓
- Outage time while instance restarted. Possibly permanent data loss. ✘

Active-Passive (MQ Multi Instance)

- Network file share or replicated block storage – latency cost ✘
- Data already synchronized to an alternative availability zone ✓
- Shorter outage while standby QM restarts ✓

Active-Active (MQ Clustering)

- Fine grain control of workload balancing across queue managers ✓
- No downtime – workload already being routed to alternative QM ✓
- Still require a strategy for restarting each separate instance ✘

Log Management

- **To manage large numbers of servers, you don't want to SSH into them very often (if ever).**

- **You will still need to diagnose problems**

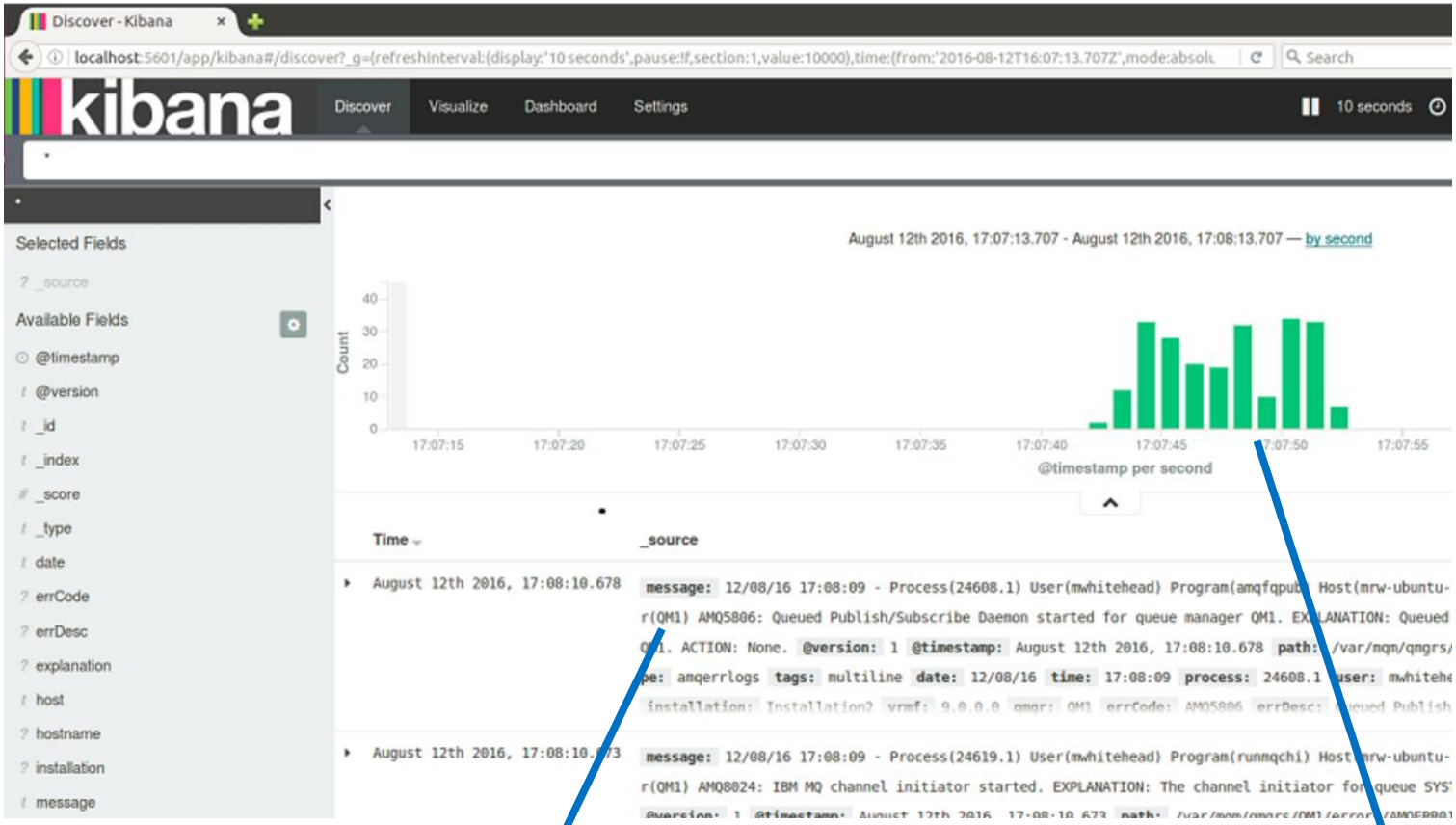
- **Centralized logging is commonly used, where an agent sends MQ and system logs to a centralized location**
  - Store
  - Index to make searchable
  - Analyze

- **For example:**
  - IBM Logmet
  - AWS Cloudwatch
  - ElasticSearch

Capturing error logs

- **In the event of a failure it can be important to gather additional diagnostics**
  - FFDCs
  - Trace
- **If you use local storage and the container or VM unexpectedly disappears you may not have access to diagnostic material**
- **If you have used networked storage do you have a way of spinning up a VM just to gather logs files?**
- **Pushing logs to a remote system (ElasticSearch, Logstash, Logmet, Cloudwatch etc.) might help separate error logs from QM runtimes, but…**
  - You may find not everything made it off-box before the VM terminated (Logstash sends every 30 seconds by default)
  - Requires scraping the file system by reading and parsing AMQERR0x.LOG

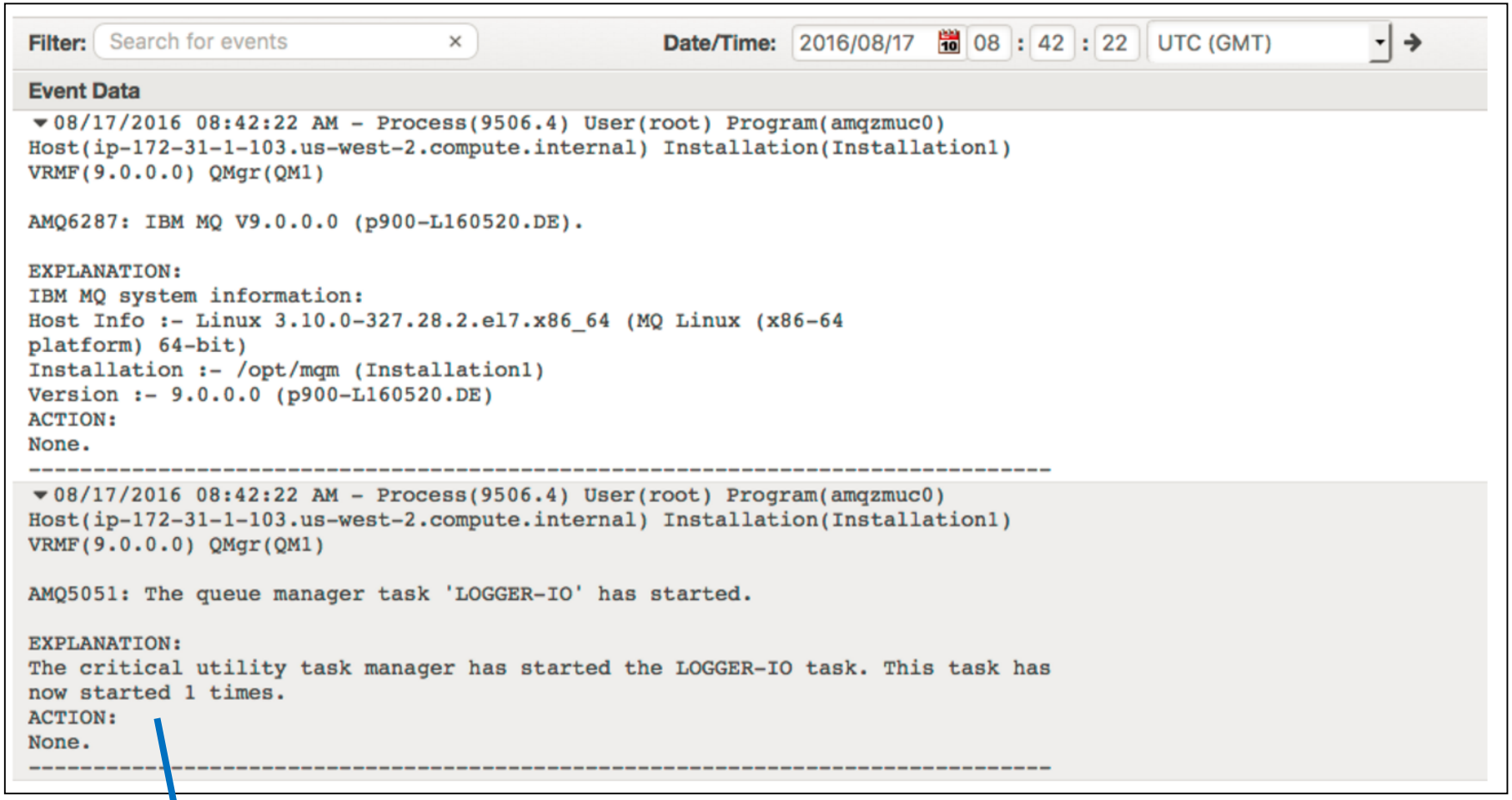
Remote Logging – Elasticsearch, Kibana, Logmet



Graphs and charts of log stats

View and drill down into each individual AMQERR0x.LOG entry

Remote Logging – AWS Cloudwatch

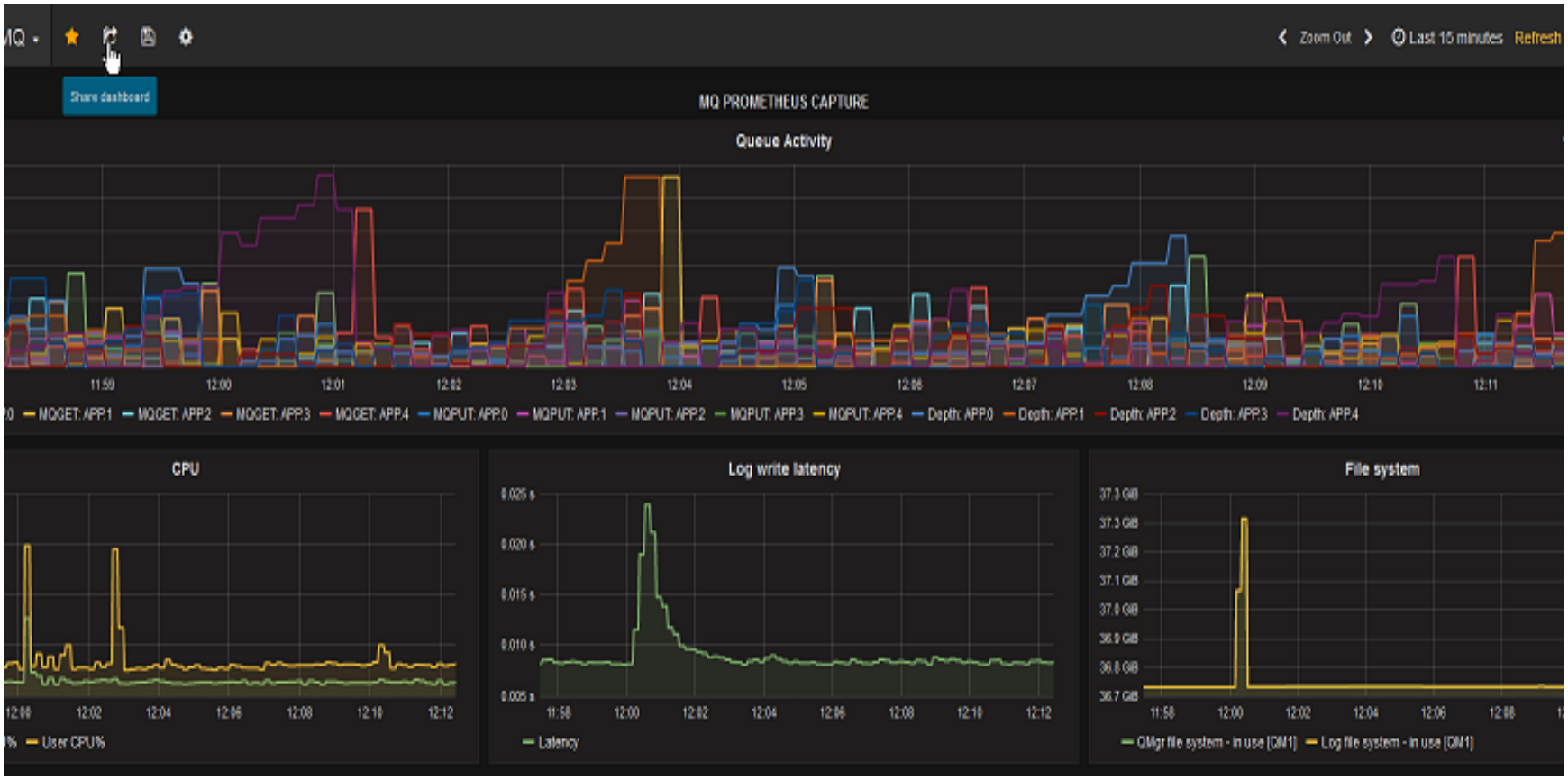


Similarly to Kibana and Elasticsearch you can apply filters and drill down into individual AMQERR0x.LOG entries

Metrics and Monitoring

- **MQ V9 makes many statistics available through a pub/sub interface**
- **Option to remotely subscribe to topics under $SYS/MQ for information on:**
  - CPU usage
  - Disk usage
  - Connections and disconnections
  - Opening and closing of queues
  - Pub/sub and put/get
  - Syncpoint calls
  - Changes to MQ objects (MQSET and MQINQ)
- **Publish to remote metrics servers e.g. Graphite, Prometheus**
  - Visualise using Grafana

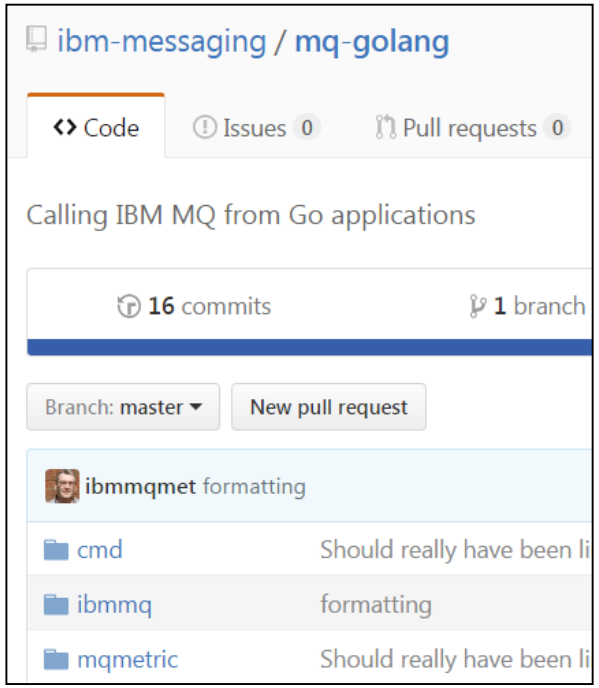
Centrally monitoring metrics



Grafana can by used as the dashboard, connecting to a back-end time series database

Mark Taylor has an MQ sample available on GitHub

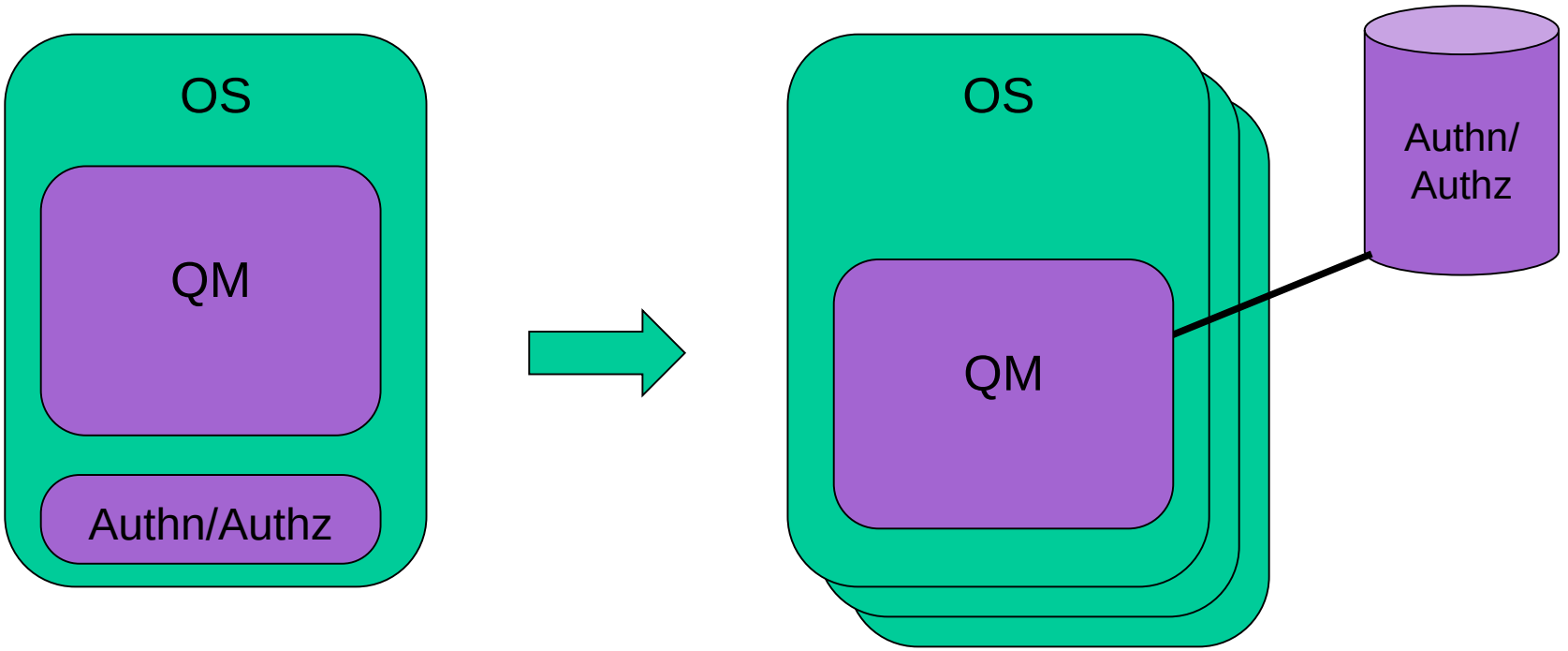
- **Written in Go**
- **Subscribes to MQ9 $SYS topics and extracts various MQ and system metrics**
- **Can be used to push data to various logging servers or databases. MQdev blog entries demonstrate:**
  - Prometheus
  - Graphic
  - Logmet
  - AWS Cloudwatch
  - InfluxDB
- **Could also output to something more generic such as collectd**



Security

- **For the large part, once you've made the choice to go with a cloud provider, the security considerations are similar to on-premise security**
- **Major differences to traditional on-premises security management:**
  - Authorisation based on OS users becoming less useful
  - Key distribution can be more challenging
  - Environment may be more dynamic
    - Need to automate as much as possible
  - Increasingly unwise to only secure at the edge of the network
    - Which was a bad practice before
- **Data isn't sat on your disks**
  - MQ Advanced Message Security can give you encryption on disk if compliance requires it

Security



- **MQ 8 Unix LDAP Authorisation (MQ 9 Windows)**
  - Blog demoing AWS Directory Service as an Active Directory repository
- **MQ 8 LDAP User ID/Password Authentication**

Security for Developers

- **Developing securely might require VPNs or secure tunnels**
- **All cloud providers off some level of VPN connectivity**
  - Typically assume some level or enterprise/appliance VPN support in the DMZ
  - Does your enterprise support the same standards?
  - Extra hoops to jump through?
- **Some offer software VPN support**
  - Well suited to developers who want to quickly connect to their cloud network
  - Vital if you don't want to expose development servers on publicly facing IP address
  - Some have limitations (Bluemix VPN currently only offers 'dial-you' mode)
  - Some have more obscure requirements
    - Microsoft Azure requires you to be using PowerShell, not the Azure CLI

Service Discovery

- **Queue manager IP addresses more likely to change in a dynamic/cloud environment**
  - Applications need to move away from hard coded connectivity details
  - Even connectivity environment variables e.g. MQSERVER=SYSTEM.DEF.SVRCONN/tcp/192.168.0.10 might not be dynamic enough
  - Consider using CCDTs, possibly served over HTTP → MQ 9
- **Most cloud providers offer manual management of IP addresses**
  - Amazon Elastic IP
  - Azure Reserved IP
  - Google Cloud Engine Reserved IP
- **DNS can provide basic service discovery for clients**
  - Amazon Route 53
  - Azure DNS
  - Google Cloud DNS

Service Discovery

- **MQ channels need to know where they're connecting to**
- **MQ clusters instances tell each other about their own IP addresses**
  - Changing the clussdr conname isn't enough
  - The clusrcvr needs changing as well, and that needs to be distributed around the cluster
  - For QM-to-QM connectivity, floating IP addresses might be a better way to achieve availbility without actually changing IP address
- **Client configuration – how do clients find the right queue manager**
  - MQ V9 CCDT URLs help centralize configuration
  - Requires an HTTP server to host the CCDTs
  - And a way of updating CCDTs when configuration changes

Blogs available on MQdev →

MQ on OpenStack, part three: Automated client connection PoC using MQ v9 CCDT URL feature.

RobParker | Aug 17 | Comment (1) | Visits (1575) 1

Cloud Ingress/Egress

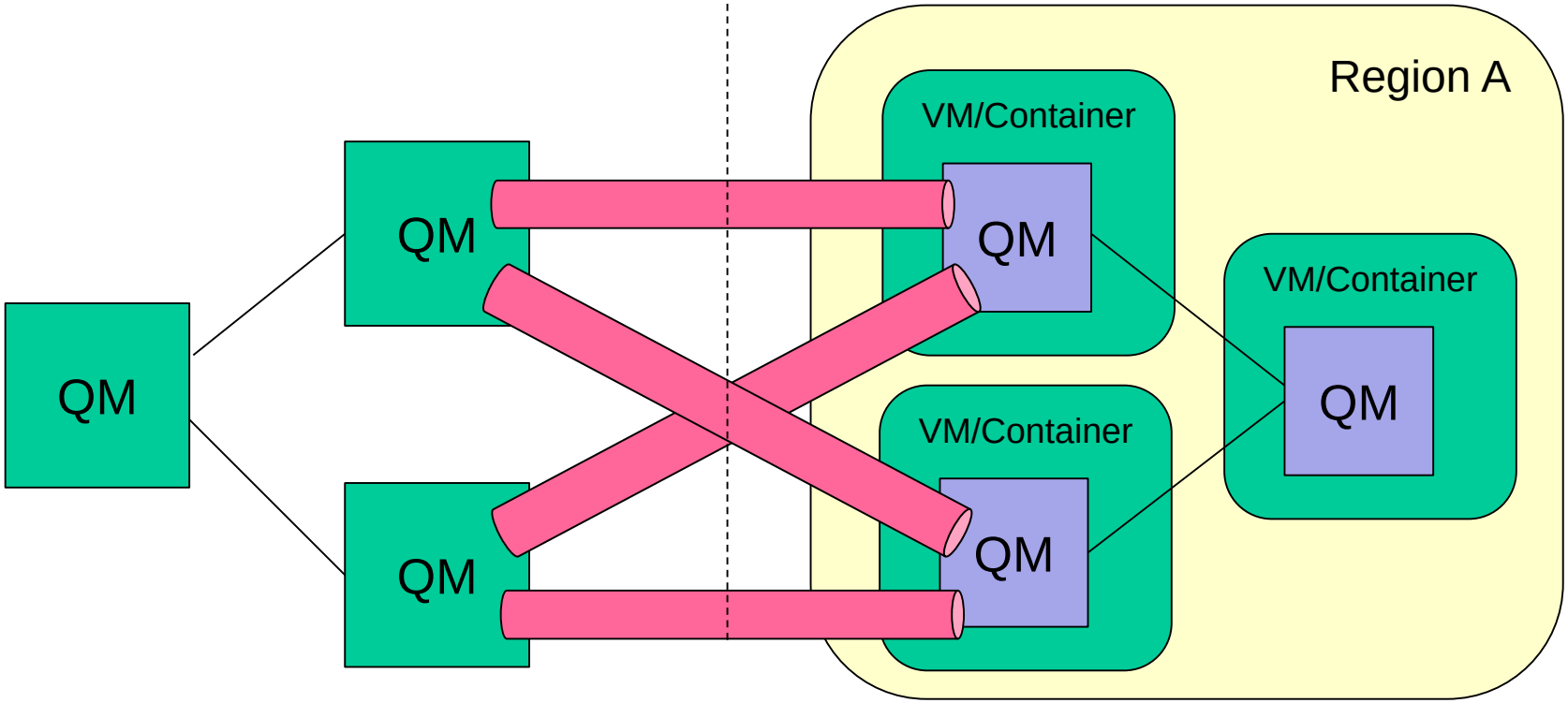
- **What about data entering/leaving the cloud?**
- **What level of availability do you need between cloud and on-premise?**
- **How is connectivity secured?**
  - VPN, Secure Tunnel, TLS/TCP
- **Performance**
  - What data rates are required?
  - Which cloud region gives the best latency?
  - Is failover to an alternative region acceptable?
- **How is data workload balanced between cloud and on-prem?**

Cloud Ingress/Egress Slide – MQ Clusters

On-Premise

Bluemix

- MQ clustering well suited to active-active round-robin of data to/from the cloud

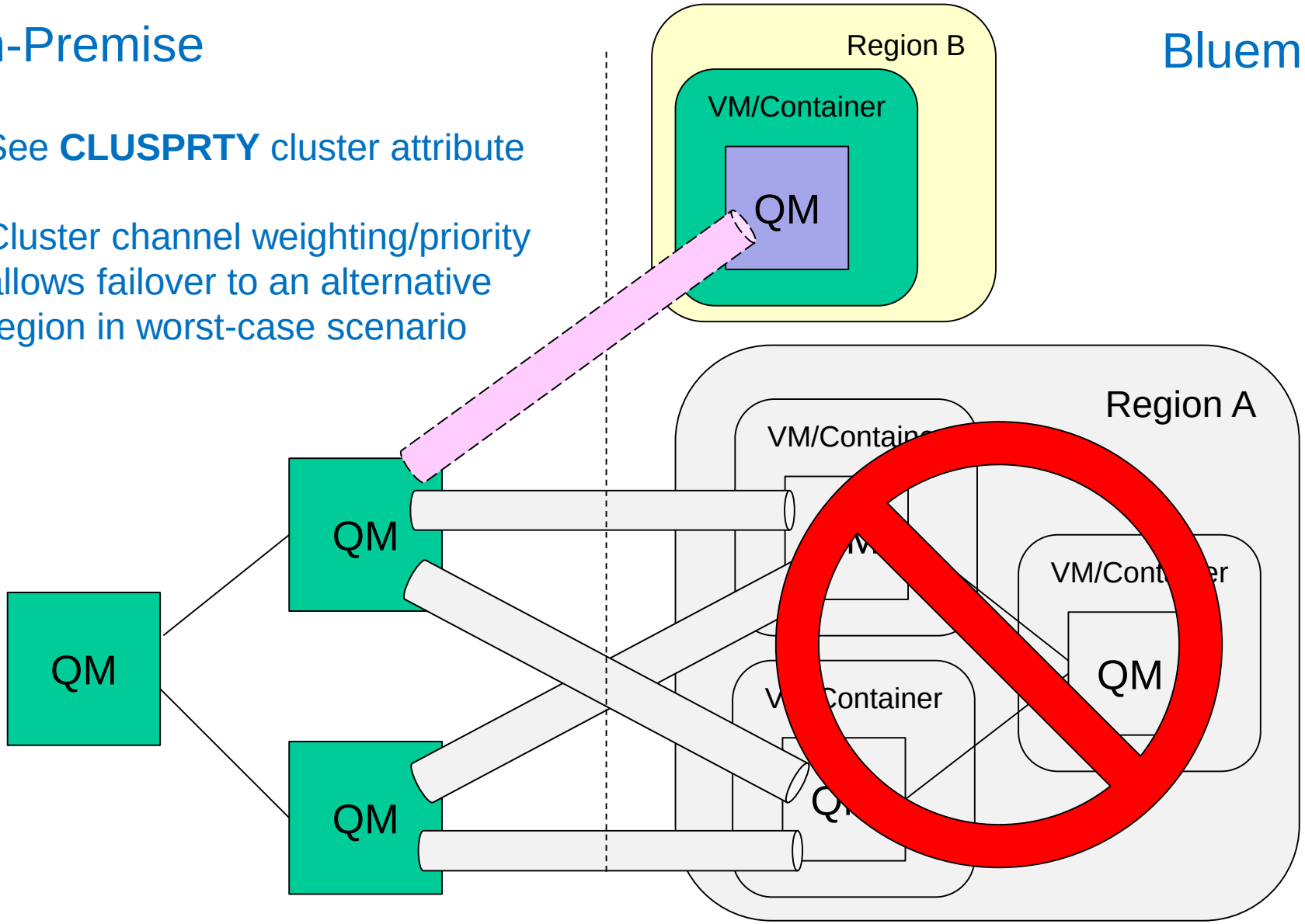


Cloud Ingress/Egress Slide – MQ Clusters

On-Premise

- See **CLUSPRTY** cluster attribute
- Cluster channel weighting/priority allows failover to an alternative region in worst-case scenario

Bluemix



MQ Ecosystem Blogs – MQdev

Using DRBD to replicate data for a queue manager

John_Colgrave | *Sep 13* | *Visits (708)* 4

I have pushed to GitHub the firs
MQ queue manager in various D

Proof of concept: MQ High availability using Ceph block storage

RobParker | *Aug 26* | *Visits (1437)* 3

In Arthur's previous proof of concept , he set up an auto scaling group using Amazon's Elastic File System (EFS) to provide a

the same AWS setup

IBM MQ - Using AWS CloudWatch to monitor queue managers

Mark E Taylor | *Aug 25* | *Visits (2530)* 6

In this final(?) blog entry of a series, I'
CloudWatch service. Previous blog en

Storing and searching MQ error logs in Elasticsearch

Matthew Whitehead | *Aug 15* | *Comments (3)* | *Visits (3683)* 6

you're not familiar with the ELK s
ributed search engine based on t

MQ on AWS: PoC of high availability using EFS

Arthur Barr | *Aug 11* | *Visits (7106)* 6

Amazon recently declared its Elastic File System (EFS) as ready for production. This enables a shared, networke
which (importantly

IBM MQ - Using Prometheus and Grafana to monitor queue managers

Mark E Taylor | *July 25* | *Visits (2383)* 3

In a previous blog entry I wrote about using the Go language with MQ. One of the reasons for creating that Go package was to enable the creation of a program that sends MQ statistics to Prometheus and hence to be easily visualised in Grafana. This blog shows how it all fits together. Introduction MQ V9 metrics MQ V9 (and the MQ appliance) makes many statistics

Thank You - Questions?

Please Note

IBM’s statements regarding its plans, directions, and intent are subject to change or withdrawal without notice at IBM’s sole discretion.

Information regarding potential future products is intended to outline our general product direction and it should not be relied on in making a purchasing decision.

The information mentioned regarding potential future products is not a commitment, promise, or legal obligation to deliver any material, code or functionality. Information about potential future products may not be incorporated into any contract. The development, release, and timing of any future features or functionality described for our products remains at our sole discretion.

Performance is based on measurements and projections using standard IBM benchmarks in a controlled environment. The actual throughput or performance that any user will experience will vary depending upon many factors, including considerations such as the amount of multiprogramming in the user’s job stream, the I/O configuration, the storage configuration, and the workload processed. Therefore, no assurance can be given that an individual user will achieve results similar to those stated here.

Trademark Statement

- IBM and the IBM logo are trademarks of International Business Machines Corporation, registered in many jurisdictions. Other marks may be trademarks or registered trademarks of their respective owners.
- Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.
- Java and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.
- Red Hat Enterprise Linux is a registered trademark of Red Hat, Inc. in the United States and other countries.
- Ubuntu and Canonical are registered trademarks of Canonical Ltd.
- SUSE and SLES are registered trademarks of SUSE LLC in the United States and other countries
- Mac and OS X are trademarks of Apple Inc., registered in the U.S. and other countries
- Other company, product and service names may be trademarks, registered marks or service marks of their respective owners.
- References in this publication to IBM products and services do not imply that IBM intends to make them available in all countries in which IBM operates.